This electronic thesis or dissertation has been downloaded from the King’s Research Portal at https://kclpure.kcl.ac.uk/portal/

Intelligent Assembly of Wind Turbine Hubs

Deters, Christian

Awarding institution:
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT

Unless another licence is stated on the immediately following page this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence. https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 04. May. 2023

Christian Deters

Intelligent Assembly of Wind Turbine Hubs

A thesis submitted in partial fulfilment of the requirements for the

Degree of Doctor in Philosophy in Computer Science

Department of Informatics, King's College London

London 2014

© 2014 Christian Deters

Abstract

The fast growing wind turbine industry is expected to play a major role in solving our energy needs in the future. At present turbine manufacturing is performed mostly manually. Due to market growth the economical peak point is reached where an automated assembly concept can be introduced.

This thesis focuses on the assembly of the wind turbine hub, in particular, on the bolt tightening process for the wind turbine bearing assembly, contributing to EU project COSMOS. Within this industrial research project, bolt tightening has been identified as an important research problem and the control strategies derived in this PhD thesis contributed to the activities of COSMOS.

A wind turbine hub has three bearings which are assembled using multiple bolts (in current wind turbines, this can be up to 128 bolts). With the need to conform to stringent safety requirements and with the aim to produce long-lasting systems, the desired clamping force between the nut and a counteracting flange needs to be accurately and reliably achieved as a result of the tightening process.

This thesis analyses the bolt tightening process divided into several tightening stages, with each stage addressing particular control and safety problems. The introduced fuzzy control architecture makes use of membership functions combined with linguistic rules to set the control target (which are specific torque and angle levels for the investigated wind turbine assembly process) to ensure that the desired clamping force is reached successfully and accurately. The control results (step response of the final control values and final clamping force) have been compared to more traditional control paradigms, including the proportional-integral-derivative (PID) controller. Experiments have shown that the accuracy improved and the standard deviation of the Fuzzy controller is more than 4 times lower than the one achieved using the PID controller

The bolt system has been further analysed and a numerical state space model has been identified using an experimental identification method. The found model has been used to identify suitable control gains for a proportional-integral (PI) control strategy and were then fine-tuned using an online learning process based on a genetic algorithm (GA).

Error detection and avoidance is another important aspect when assembling safety-critical systems such as wind turbines. This PhD study introduces an error detection mechanism that is active during the bolt tightening process and integrated with the fuzzy control architecture used for bolt tightening. This is achieved by defining additional membership functions and linguistic rules for error detection. The error detection mechanism is based on a logic based approach terminating the tightening process when critical control parameters are exceeded.

Contents

List of Tables

h

Nomenclature

FLC	Fuzzy Logic Controller
LT	Low Torque
NT	Normal Torque
HT	High Torque
AL	Angle Low
AN	Angle Normal
AH	Angle High
VN	Voltage Negative
VP	Voltage Positive
VZ	Voltage Zero
RT	Tension limit reached
CL	Close to Tension limit
BE	Below Tension limit
PLC	Programmable Logic Controller
TC	TwinCAT
MAMD	Mamdani Fuzzy Controller
A_i	Fuzzy Parameter
B_i	Fuzzy Parameter
P	Proportional Controller
I	Integral Controller
D	Derivative Controller

PI	Proportional-Integral Controller
PID	Proportional-Integral-Derivative Controller
F	False
T	True
TL	Tension limit
TIGH	Tightening active

Acknowledgements

I would like to express my deepest gratitude to my supervisors, my colleagues and my family. This thesis would not have been possible without their guidance and support.

I would first of all like to thank my supervisors, Dr H. K. Lam and Prof Kaspar Althoefer, without whom I would not have had the chance to embark on, conduct and complete my PhD. I am grateful for their suggestions, encouragement and for making my experience in King's College London an enjoyable one.

In addition, I am indebted to all my colleagues from the Centre of Robotics Research who supported me throughout my research, sharing their thoughts on various technical discussions. I would like to thank especially Emanuele Lindo Secco, Asghar Ataollahi, Ramon Sargeant, Dinusha Zbyszewski, Jim Trotter and Vahid Aminzadeh.

Furthermore, I would like to thank the Department of Informatics of King's College London and the EPSRC for arranging departmental funding for my PhD.

Thesis Outline

Chapter 1 – The first chapter provides an introduction to the wind turbine industry and the issues encountered in the wind turbine hub assembly process. More specifically, the bolt tightening process which is used in the assembly of the wind turbine hub. The current bolt tightening processes do not provide sufficient accuracy, which motivated the development of the two intelligent assembly strategies introduced in this thesis.

Chapter 2 – "Review of tightening tools and control strategies" investigates and contrasts manual tightening tools as well as automated tightening tools used in industrial settings. Furthermore, various control strategies have been analysed and compared in the context of bolt tightening applications, identifying strategies suitable for bolt tightening in wind turbine manufacturing.

Chapter 3 – "Bolt tightening based on fuzzy control for wind turbine bearing assembly" – describes a bolt tightening controller based on a model-free fuzzy control strategy. The control system described is based on the assembly strategy and includes the pick and place process, the alignment of the nut on the bolt, the run down phase and the tightening. Error analysis is part of the study.

Chapter 4 – "Model-based self-tuning PI control of bolt-nut tightening for wind turbine bearing assembly" - introduces a self-tuning PI Controller using an online learning algorithm. An approximated numerical model of the bolt system has been derived and used to estimate the PI gains. These gains have been used in the PI controller as part of the experimental study on bolt tightening; it could be shown that fine-tuning the gains with an online genetic algorithm improves the controller performance considerably.

Chapter 5 – "Comparisons of the control concepts" - compares the results of the model-free fuzzy controller concept introduced in chapter three to a model-based self–tuning fuzzy controller based on a Genetic Algorithm. Comparisons are performed with regards to reliability, system response and error recognition.

Chapter 6 – In this chapter an outlook is taken on an intelligent adaptive concept for the automated assembly of the wind turbine hub. An agent-based approach has been used to realise a search algorithm which creates a wind turbine hub assembly strategy.

Chapter 7 – This chapter provides a summary of the conducted research and of the research contributions is presented. Possible future work involves merging the control strategies into the agent based concept where the wind turbine hub is assembled in a flexible adaptive manufacturing environment.

Appendix A – This appendix provides a more detailed technical background on the developed test bench MATLAB Simulink models and how they are implemented in a real time control environment. The workings of the fuzzy controller are explained as well the way the switching between the stages is implemented.

Appendix B – This appendix describes the designed clamp system which prevents backward torque into the robot arm. A bolt bench has been designed to implement a pick and place process with a robot arm with the connected tightening tool performing the complete tightening process. Furthermore, another bolt bench which is used for several experiments is described. This bench also allows examining the performance when applying control strategies during fault scenarios, as it is able to accept a high torque. The bench is equipped with a clamping force sensor to measure the final clamping force.

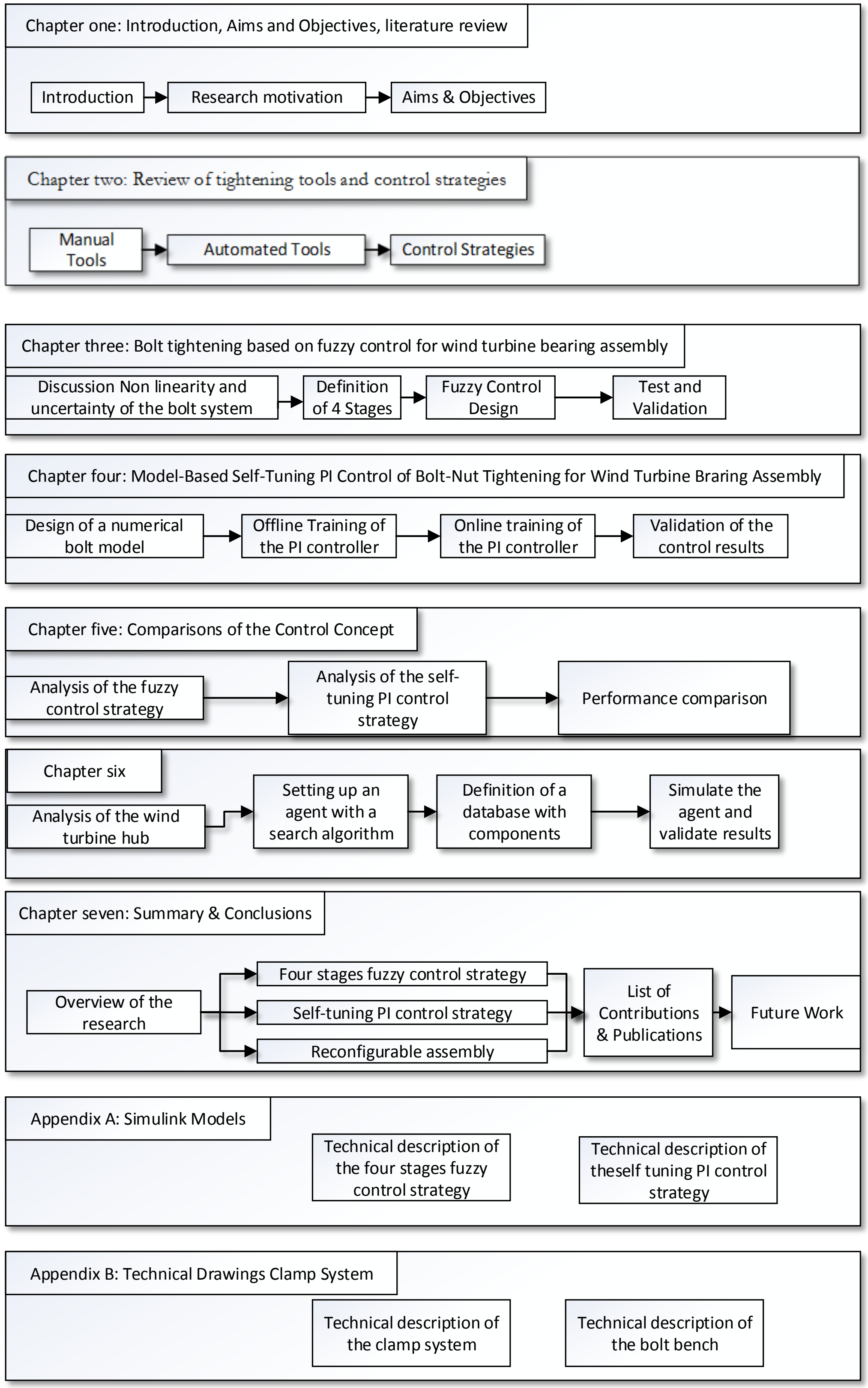


Figure 1: Thesis structure

CHAPTER 1

1. Chapter 1: Introduction

1.1. Wind turbine industry: An Overview

Traditional energy sources such as fossil fuels will run low in the future and it is essential to research and invest in alternative renewable sources, such as solar energy and wind energy. The wind turbine industry has been growing over the last 20 years and wind generation capacity has reached a worldwide capacity of 93.8GW. As more countries, such as Germany, the UK and the United States, are increasingly investing into wind energy, the industry is expected to further expand in the near future [1]. Currently, the demand for more powerful turbines both onshore and offshore is increasing, and it is likely that it will continue to do so over the next years especially in countries with comparatively lower sunshine amount [1].

Wind energy has been used to create electricity for decades and the technology has matured to the point where a single wind turbine can reach a power output of up to 7MW. Wind energy is expected to provide 12% of the world's overall energy requirements and the research and development on wind turbines focuses on making the turbines even more powerful [2] [3] [4] [5].

Wind turbines have been assembled mostly manually as low demand and output in the past had made the introduction of an automated assembly strategy financially unfeasible. However, with demand peaking nowadays the introduction of automation in wind turbine manufacturing is becoming economically interesting.

A wind turbine comprises a wind turbine hub, a tower, and a nacelle, as shown in Figure 2. Depending on the turbine manufacturer, the tower contains the power converters for connecting the turbine to the grid, the nacelle contains the generator and the gearbox connected to the rotor and the hub carries the three rotor blades as well as the electrical pitch system with all its components.

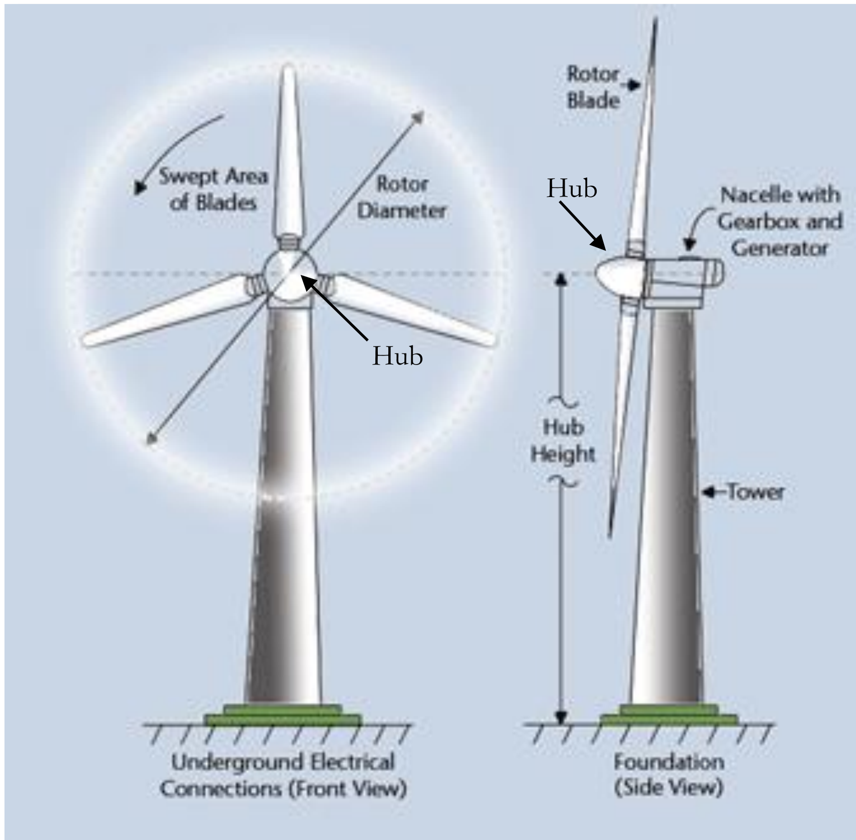


Figure 2: Wind turbine structure [4]

Wind energy has been used for over 3000 years [5] for sailing the seas, pumping water and grinding grain. The 3 axis horizontal wind mill was designed in the 13th century and mills played a major role in the early economy before the introduction and widespread use of fossil fuel. As early as the 19th century a 12kW wind mill was used to produce electricity. However, as the electricity grid was further developed interest in wind energy remained fairly low until 1941, when Smith Putham [5] designed a wind turbine with a power output of 1250kW. This machine was using a steel rotor with a radius of 26.5m. More wind turbines have been developed afterwards with a much better energy outcome but despite the advantages there was only little interest for this type of energy generation until 1973, when the oil price rose dramatically. As a result, governments started looking for alternatives and they set up research funding, which increased the general interest in wind energy. However, at this time, there was a high uncertainty about the design of cost efficient and energy productive wind turbines. Several different types of wind turbines have been developed and their efficiency was analysed and compared. The wind turbine design that was developed in the early 1980s is the one which is still common nowadays. At that time, there was still an uncertainty about the number of rotor blades a wind turbines should have (there are designs for 1, 2 or 3 blades [5]). Moreover, turbines could not run unsupervised in an offshore environment, for example damages could not be identified at an early stage (for example in the gearbox) which lead to bigger damages. In the worst cases the whole mechanical set up of the wind turbine was destroyed. It was also a problem to keep the turbines running on a constant rotation speed at different wind speeds. If the wind speed got too high, the turbine could get damaged, if it was too low, the turbine could not start to work. To address this, several different concepts were developed, including the so-called Danish wind concept, which includes a three blades rotor (Figure 2), and is stall regulated to keep a fixed rotation speed. This design has been found to work very efficiently and therefore it is successfully used today.

The demand for wind turbines is further increasing and it is estimated for example that in Denmark wind energy currently provides 20% of the overall energy production and the plans are to push it up to 50% [1]. Furthermore, the demand for offshore wind turbines increases as the wind distribution on the sea is more constant than on land including predictability of the wind direction. Compared to onshore turbines, the demand is lower [6], since the usable area for offshore wind turbines is smaller (only available at certain coastlines). Furthermore, the infrastructure for offshore wind turbines is more complex which causes an increased investment for setting up the turbines [2] [4] [5] [6] [7] [8] [9] [10] [11] [12].

1.2. Aims and Objectives

The main aim of this thesis is to design an intelligent bolt tightening strategy for wind turbine hub assembly to potentially replace currently used manual installation processes in the wind turbine industry [9]. Furthermore extending on this strategy a proposed agent-based method promises to achieve automated assembly of the wind turbine hub.

The proposed bolt tightening strategy aims to ensure expedite assembly according to specifications, and more importantly, to achieve a high-standard tightening process. Therefore a more time consuming control approach which accomplishes the tightening process on a higher standard is preferred to one which may be faster overall but has a higher distribution in the final torque and angle level.

To this aim the thesis will proceed with:

A. An in-depth analysis of the bolt tightening process

This involves an account of problematic phases and division of the process into different stages, capable of coping with inherent nonlinearities and uncertainties, such as those due to variations in friction and material properties. Possible bolt tightening errors, such as a misalignment of the nut on the bolt or different bolt-nut size combinations are analysed and classified according to the type of error.

B. The development of two approaches to bolt tightening:

a) A model-free approach based on fuzzy control using a model-free Mamdani-type fuzzy controller, and

b) A model based approach combined with an online learning Genetic Algorithm (GA) for fine-tuning.

In approach a) the fuzzy controller is based on the torque/angle tightening technique and employs several controllers to integrate all aspects of the system. Furthermore, the controllers are running the tightening tool on different speeds to prevent any physical damage to the system. A lower speed will decrease the kinetic energy (which can damage the bolt system in an error).

In approach b) a numerical model for offline training is determined. A genetic algorithm fine-tunes the originally derived gains of the controller until the step response of the system is improved. This approach allows coping with bolt systems nonlinearity and uncertainties.

C. An investigation into how the bolt tightening strategy can be extended to an agent based concept for automated assembly of the wind turbine hub:

In this investigation, the assembly of the hub has been generalized and the assembly of the wind turbine hub has been simulated by determining the assembly requirements of the main components (hub body, pitch bearings and bolt connections). A search algorithm has been selected (Partial Order planning, POP) and the assembly strategy has been derived in a simulation environment.

All approaches are implemented on industrial computer hardware in the form of hard real time applications, to demonstrate their suitability to industrial environments.

1.3 Research Motivation

Almost all turbine components – electrical components, wind turbine nacelle, and turbine hub - are assembled manually in the factory. The final part of the assembly process takes place at the customer installation site where the nacelle is placed on top of the tower and the hub is installed on the turbine's nacelle.

In this thesis research focuses on the pre-assembly stage in the factory and in particular on the bolt tightening process which is one of the core processes. The proposed tightening strategies are expanded to an agent-based automation concept which is introduced in this thesis [10] [11] [14] [16] [17] [18].

The bolt tightening process of the wind turbine hub is currently performed manually using torque wrenches and hydraulic tensioning tools, such as introduced in Chapter 2. The main target of a bolt connection is to create a specific clamping force. The clamping force is the force applied between the bolt head and the nut and a product of either the final tensioning force (if the tensioning approach is used) or the final torque and angle levels (if the tightening approach is used).

Figure 3: Wind turbine hub with placed bearings [12]

Figure 3 shows the pre-assembled wind turbine hub with installed bearings in a factory setting. Several bolts can be seen arranged in a circle mounted on the bearing on the wind turbine hub. The bolt connections which attach the bearing to the hub can be seen on the outer bolt ring; the inner ring (the holes around the bearing in Figure 3) will be assembled on site and mounted on the turbine blade on the hub once the latter is mounted in turn on the nacelle.

The number of bolts connecting the bearing to the hub varies but there can be up to 128 bolts of different sizes. When the turbine runs, the load of the wind pushes against the blade causing it to rotate. The wind load is also applied to the bearing and at the wind side (i.e. the front side of the turbine) it may put excessive pressure on the installed bolt connections. These bolts are the only means whereby the 50-metre long blades are linked to the hub and through the hub to the gearbox and the generator. If the tightening process is not completed accurately and the required clamping force is not reached or is higher than specified, a local overload of individual bolts may cause the bolt connections to fail during operation.

This failure could cause a shutdown of the turbine and could necessitate the replacement of the bolt connection. Depending on the level of damage, it may be necessary to take the turbine blades and hub down to conduct repairs. Furthermore, if several bolt connections break, a blade may hit the tower as it bends back. This may cause irreparable damage to the turbine and/or it may be required to repair the turbine – it is noted that during repair, the turbine is not producing energy.

It is essential that the wind turbine is assembled to the required assembly specification in order to avoid failure, turbine damage and ensure public safety. For example, a blade failure can throw parts of a wind turbine blade up to 2km away from the turbine [13].

The following Figure 4 shows damages, which can cause a public health risk; such damage could be caused by an inaccurate assembly process. It is an example of irreparable damage and a risk for public health as turbine parts can fly uncontrollably and can land quite some distance away from the turbine.

Figure 4: Damaged wind turbine blades, can cause public health risk [14]

The example in Figure 4 shows serious damage where a replacement of the turbine may be necessary. The damage may involve a long downtime, reduced energy provision and reduced financial income; also replacing the turbine is rather costly. In particular, sites which are difficult to reach (bad infrastructure) or offshore turbines can lead to rather high economic loss.

Thus, it is essential to detect errors during the tightening process, to ensure that the assembly is conducted to specifications; concerning the clamping force in this context, it is imperative that the appropriate clamping force is reached on all bolts across the entire bearing. The bolts need to be properly installed and must not be affected by any mechanical damage.

An error recognition system must be able to actively monitor the tightening process, ensuring that the correct bolt is installed, the appropriate forces are achieved and ceasing operations as soon it detects an error.

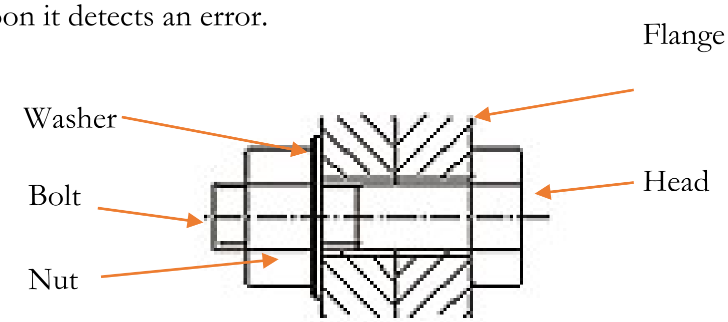


Figure 5: Bolt connection [15]

Figure 5 shows a schematic of a bolt connection. The bolt itself is a rather complicated element, even though it appears to be simple. A bolt is actually a spring [16] [17] – it is stretched during the tightening process and locked in a particular position using a nut. There are many different types of bolts and manufacturing standards [18] which makes the actual tightening process different for various bolts [19].

The standards vary all over the world as different industrial applications use different bolt types. However, in the wind turbine industry, mainly metric bolts are used [9].

There are many different tightening techniques which can be used for the completion of the tightening process, such as the torque or angle technique. Both individual tightening techniques can be combined [16] [17]. Each technique requires an individual controller; when both tightening techniques are combined then both controller outputs will be added, as described in Appendix A. This allows to tighten the nut to a specific torque as well as a specific angle level.

The tightening process is affected by twisting due to the rotational movement of the nut on the bolt while the friction level between the nut and the bolt influence torque level. The friction level is varying and depends on the material properties and on the corrosion protection, in particular, if the bolt is oiled. The clamping force will drastically change if the friction level varies even though the same tightening torque has been applied.

The assembly process of the wind turbine comprises several steps. One of the core assembly processes is the bolt/nut connection between the bearing and the wind turbine hub. This process is currently done manually by using different tools such as torque wrenches, hydraulic tensioning tools and gauges [20]. In addition, the bolt nut connections need to be assembled to a high standard to ensure that the desired torque angle tightening level of each bolt meets the assembly specification. It has to be ensured that bolts are tightened on the specified level and – if not – the faulty level needs to be detected as well as errors which prevent the tightening process or may cause damage [10] [14] [16] [17] [18] [21].

Current automation technologies used for bolt tightening processes are mostly based on PID controllers which can be used for the torque angle tightening technique [16]. PID is in general well accepted in industry and its performance is accurate in particular on linear systems. A bolt is acting as a spring in hydraulic tensioning systems, the performance is very good but it decreases when non-linearity and uncertainties are introduced to the system. The tightening process introduces many uncertainties and non-linearity, such as mechanical frictions between the threads and the head of the bolt and the surface of the flange, uncertainties [22] are also introduced by variations in temperature, variations of the friction level (e.g. an oiled bolt will have a significant lower friction level) and the presence of possible damages which may affect the performance of a PID controller and make the bolt system highly non-linear [23] [24] [25] [26] [27]. Furthermore, a PID controller with fixed PID values may not be able to comply with the performance requirements hence it cannot integrate the non-linearity and uncertainty of the bolt system.

Thus, alternative control strategies which can integrate the non-linearity and the uncertainties are needed, since PID is an approach which requires accurate information about the control system (and the performance varies due to the uncertainties and non-linearity) [28]. The use of a model free fuzzy controller can be used to integrate all the quality control requirements for this application. Also the control strategy can be used on several different bolt sizes since it is not based on a theoretical numerical model describing the system attributes.

Another essential aspect is the error detection functionality. Common errors, such as misalignments of the nut on top of the bolt have been defined and need to be detected.

A theoretical approach for screw fastening has been introduced in [29] to address non-linear components of the system which can be controlled using a fuzzy controller. However, this approach may not be used for this application, as the hub assembly requires higher accuracy and error recognition.

Also, industrial integration has not been targeted in the screw insertion application [29]; different rotational speeds have not been considered to prevent mechanical damages in critical areas of the tightening process. The approach introduced in this chapter allows tightening of bolts on a high quality level including quality control feature as well as damage prevention during the process.

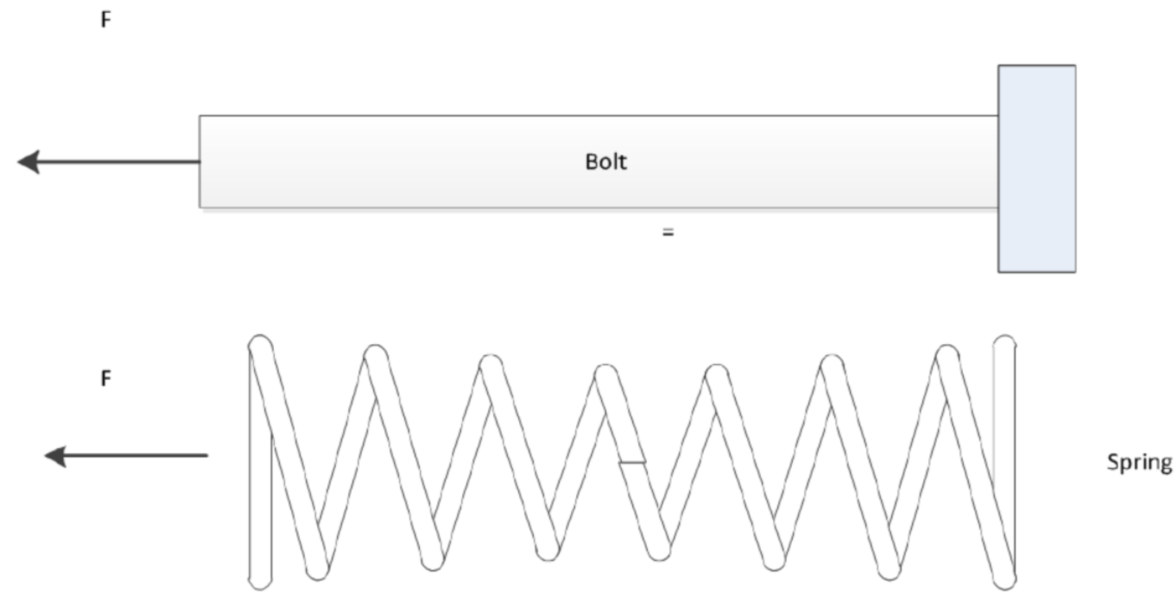


Figure 6: Bolt Spring system

The bolt tightening process using a robot will use a robot arm for the pick and place process of the nut and will place it on top of the bolt of the wind turbine hub. Once the tightening is completed, it has to be ensured that the final reached clamping force is according to the desired specifications and that as a result the nut is not becoming loose during operation of the turbine [22] [30] [31] [37] [38] [39].

In order to complete the tightening process, a control strategy needs to be chosen and implemented in the tightening controller. The tightening controller can be either a classical model-based controller estimated on the characteristics of the bolt model or a model-free control approach, which experimentally showed a higher accuracy and reliability than a model-based approach. A bolt connection contains non-linearity and uncertainties which make it difficult to estimate a general numerical bolt model which can be used to develop a bolt tightening control strategy.

Even though bolts and nuts are mass products and standardized and the bolt has a specific thread type (in this thesis only metric bolts are considered, as only metric bolts are used for wind turbines [9]), they suffer from some material variations. These variations are possibly even higher when the bolts and nuts are made from different manufactures. In addition, bolts and nuts can be oiled before they leave the factory to prevent oxidation and ensure an easy installation. However, this means that the friction level of the bolts and nuts varies and that a different torque and angle level may be required for the tightening process in order to reach a specific torque level [16]. Other uncertainties can be caused by variations in storage conditions. Both a cold and dry warehouse environment as well as hot and wet storage conditions can affect friction levels of the bolts and nuts.

The way the bolt is preinstalled by the robot adds another uncertainty, as the tightening angle changes; the washer installed on the bolt which has also variations in the material and surface friction like the bolt and the hub flange. When the nut runs down the bolt it is affected by variations of friction on the bolt and when it reaches the washer and the flange, the friction is a combination of the thread, washer and flange friction. The overall system is also affected by the temperature, as this changes the material properties and friction levels.

In summary, all these uncertainties make it rather difficult to estimate an accurate numerical bolt model which can be used to design a precise controller. To this aim, this thesis proposes the development of a model-free control strategy which will overcome the described non-linearity and uncertainties and guarantee an accurate result. Fuzzy control can be implemented with or without a model and overcomes nonlinearity and uncertainties in the bolt tightening process [23] [25] [29] [30] [31] [32] [37] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59]. Fuzzy control strategies can be used in this application as it can be set up using expert knowledge about the bolt tightening process. Using membership functions, the inputs of the system, such as the angular and torque feedback, can be set and the desired torque and angle levels can be defined and linked using linguistic rules. This approach can potentially provide a constant quality level even though the bolt system changes its attributes.

Alternatively a model based approach can be used. In this approach a numerical model needs to be identified either theoretically by describing the physical system with all its components (namely the tightening tool with its servo-drive, its gearbox, its spring mounted shaft and the bolt system containing the bolt, the nut, the washer and the flange) or experimentally by combining several step response data and identifying a numerical system. Several model types can be used, such as Hammerstein-Wiener models, neural networks or state-space models. For this research, a state-space model has been chosen as it showed the best performance compared to other model types. The approach chosen here to realise the numerical model makes use of the MATLAB system identification toolbox.

However, the numerical model will not show all the nonlinear characteristics and uncertainties of the bolt system. The model will show sufficient accuracy for the initial design of a control strategy to reach the control target, but the controller performance may not be optimal.

The tightening process can be based on a PI control strategy using gains which can be fine-tuned. A genetic algorithm can then be implemented and combined with the PI controller and set up with the initial derived gains. These gains are then fine-tuned online during each tightening process until the control performance reaches the best possible level.

For the model-based tightening strategy, the angle based tightening strategy has been chosen [16]. It has the advantage that it will not be directly influenced by the uncertainty and non-linearity of the friction level as the nut runs to a desired target angle. A tightening angle should always reach a particular clamping force. However, the angles change due to variations in the material properties, as the starting position of the thread (that is the nut grabs into the bolts thread and starts the run down process) changes with different bolts. The starting position also depends on how the operator pre-assembles the bolt. Therefore, each bolt will then have a slightly different starting angle which means that the final angle will vary by a few degrees. This introduces an uncertainty to the final clamping force of the bolt connection.

This issue may be overcome by measuring the torque level during the run down process; the nut runs down until it touches the flange and at that moment the torque level increases. This position can be seen as a zero position and the tightening based on the angle technique can be completed.

The uncertainty which arises employing the above technique is due to torque measurements during the tightening process. However, for this approach to work the torque that occurs when the nut hits the flange needs to be clearly discernible when compared to the run down torque. During the run down the tool has to overcome the friction level of the bolt, but the torque associated with the run-down of the nut is usually considerably lower than the torque needed to overcome the friction between nut and flange in the final stage of the bolt tightening process. This approach to define the zero position showed to be more reliable in experiments (than the above angle based strategy), as the mean value distribution of the zero angle turned out be lower compared to the zero position when the nut is placed on top of the bolt.

Errors may occur during the bolt tightening process and it has to be ensured that the nut is tightened according to the assembly specifications. If for some reason that is not possible, e.g. due to a mechanical defect, the tightening process has to be stopped and the bolt or the nut has to be replaced. Furthermore, the occurrence of an error situation should not need lead to further mechanical damages of the bolt or even the hub system. An exchange of the whole bolt nut connection may well be avoided as it may possible to replace only the faulty part. This fault detection process for automated assembly has been analysed for screws in [33]. The screw insertion process has different error sources and also a different installation method. Another method described in [34] uses an electric screwdriver with a torque sensor and an encoder. This approach uses torque signatures to detect any failures based on neural networks [35].

Other researchers have provided solutions for decision making algorithms employing a Mamdani Fuzzy Controller using fuzzy rules and membership functions [36] for implementing which can be used as a base for both the tightening controller as well as the error detector. For the model-based strategy based on a PI controller a separate program can be developed which monitors the torque and angle levels.

The process needs to run in an industrial automation environment using a robot. Both the robot and the tool need to be connected to the controller as well as to the rest of the assembly line, ideally via a modern and reliable fieldbus system. Furthermore, several tightening tools may be used for the assembly process and they may need to work simultaneously [37] [38].

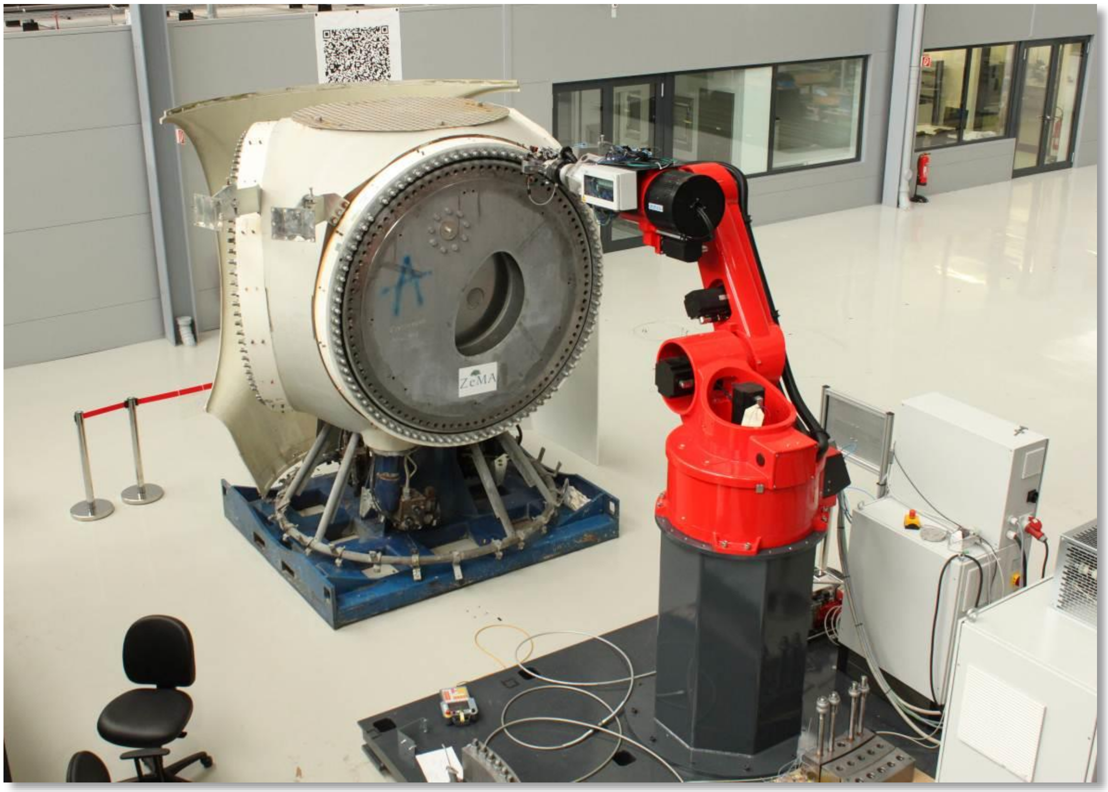


Figure 7: Robot assembling the bearing of a wind turbine hub (provided within the COSMOS project)

The picture above shows the assembly of a bearing on a wind turbine hub body. The picture shows a validation set up used in the COSMOS project, to which this PhD study contributed to.

Figure 8 shows the hub when it is finally installed in the field. All three bearings are installed. The figure also shows that corrections to the existing bolt connections causes a more complicated process, as the hub is on a height of up to 120m.

Figure 8: Installation of the wind turbine hub in the field [39]

CHAPTER 2

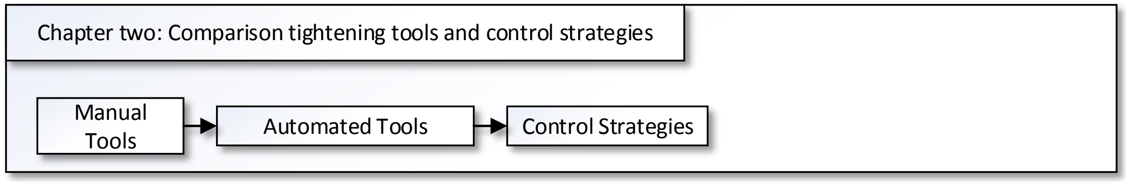
2. Chapter 2: Review of tightening tools and overview about possible control strategies

2.1 Chapter outline

The wind turbine assembly contains many core processes. One of the core processes is the assembly of the wind turbine hub; in particular, the assembly of the bearings of the wind turbine hub.

This is currently done manually, using torque wrenches or hydraulic tensioning tools. This chapter provides an overview of the current manual assembly tools and also provides an overview of tools which can be used for the automated wind turbine assembly application.

Another aspect in this chapter is the review of control strategies for the bolt tightening application. The bolt system is a nonlinear system including uncertainties and non-linearity making the bolt system a challenging control application; this chapter analyses suitable control strategies.



2.2 Overview of industrial tightening tools

The bolt system is a mechanical connection using a bolt, washers and a nut. In between the threads of the bolt and the nut as well as between the bolt head, the washers there is friction. The friction level may vary depending on the condition of the bolt (e.g. corrosion, oiled bolts, variations in material, variations in the bolt stiffness etc.) Another aspect is fault detection. Faults during assembly can be caused by using a faulty bolt.

There are many different tightening tools, the most suitable ones for this application which will be introduced in this chapter.

2.2.1 Torque wrench

One of the simplest tightening techniques is based on a mechanical torque wrench. The wrench can be set to a particular torque value set in [Nm] and the nut is pre-installed on the bolt and run down to the flange. By rotating the wrench, torque is applied to the bolt system until the final desired torque level is reached. The wrench will then mechanically open at the desired final torque.

Figure 9: Mechanical Torque Wrench [40]

Another manual approach is based on an electronic torque wrench. Compared to the mechanical wrench which is using a spring system to detect the desired torque this system uses an electronic system based on a strain gauge torque sensor. By using this electronic sensor a higher accuracy can be achieved, as the accuracy of this sensor is higher.

Figure 10: Electronic Torque Wrench [41]

It needs to be considered that the introduced torque wrenches do not have an integrated control strategy. They will only mechanically (triggered electronically) disengage when the final torque is reached. If the tightening is done very fast they may be a torque overshoot which decreases the final accuracy and introduces an uncertainty to the final clamping force. However, if the tightening is done slowly, a high accuracy can be reached with the electronic torque wrench.

The manual tightening process has also the advantage that errors can be detected easily by the operator during assembly as he or she monitors all the assembly steps.

Figure 11: DSM Messtechnik manual tightening tool [42]

2.2.2 Automated tightening tools

The bolt tightening process is already automated in many fields (e.g. automotive industry). There are already automated tools which are used for the automated bolt tightening process. There are digital tools and analogue tools. The next figure shows an automated tightening tool, which has already an integrated industrial bus system (such as ProfiNET). The tools are based on linear control strategies, such as PID.

Figure 12: Industrial automated tightening tool [42]

It is noted that there are two different types of tightening tools. The most used ones are a combination of a tightening tool connected to an assembly station and a transducer with an integrated controller and an industrial interface to set torque/angle values and further tool parameters. Other ones, older tools, are analogue tools, which are used in combination with a PLC running the control algorithm.

With regards to accuracy, automated tightening tools reach up to 0.5% from the final torque and 0.1° for the angle. This is the maximum achievable accuracy, based on the accuracy of the sensors, e.g. the strain gauge sensor used for measuring the torque and the encoder measuring the tightening angle.

2.3 Control Strategies which can be used for non-linear control systems including uncertainties

There are many control strategies which perform well on particular control applications, such as PID controllers for linear controllers. Linear control strategies lack the capability to adapt to non-linearity.

Another requirement for a control strategy is that it can be used in a control environment, where discrete controllers can be implemented (such as PLC systems).

Furthermore, the bolt connection is using a tightening tool with two outputs, namely the torque and the angle. Therefore, a control strategy needs to be selected according to this requirement, i.e. being capable of operating with multiple input signals.

2.3.1 Predictive Control strategies

There are already various controllers which have been described in current research publications, such as a torque controller based on a predictive control strategy [43], which is able to adapt the systems behaviour and adapt to system changes based on predictive methods. However, the accuracy of the prediction may not be high enough to be used on a physical system, such as a bolt tightening system, as it suffers from non-linearity and uncertainties, which are difficult to predict.

2.3.2 PID for tensioning tools

Another bolt assembly strategy is based on bolt tensioning. This strategy is different from to the tightening strategy, as the bolt is pre-tensioned and then the bolt connection is established. This approach is also non-linear, but can be represented by linear models more readily and uncertainties are significantly less when compared to bolt tightening.

The tensioning approach can be based on a classical PID controller, as the bolt system will act over a wide range as a spring, i.e. with a fairly linear behaviour. Non-linearity occurring in bolt tensioning can be usually neglected. For example, the process described in [44] reporting on hydraulic tensioning can be accurately completed using a standard PID controller, although some non-linearity is observable. However, and as this PhD thesis will show, assembly processes based on bolt tightening can be considerably improved with regards to accuracy employing intelligent control strategies capable of coping with high non-linearity and uncertainties.

2.3.3 Neural Network Controller

Neural Network Control [45] is a biologically inspired control approach mimicking some function of the brain. This control strategy can be applied to nonlinear dynamic control problems, such as the discussed bolt-tightening problem. In particular, a neural network is able to learn the systems attributes even on a poorly modelled nonlinear dynamic system. However, it may not be necessary for a linear control system or a system which can be linearized.

In an industrial environment a neural network is used in different scenarios [45], such as in combination with a database of correct control signals which are used to train the neural network to adapt the correct signals. Although good results can be achieved with neural networks, the main issue that remains is that it is difficult to understand what exactly they have learned. Hence, they may give good responses for a subset of the overall range of operations, but may fail to operate appropriately in other parts. Because of this general issue, they are also referred to as "black boxes" and certifying them for industrial use can be problematic.

Another strategy is called indirect adaptive control where a neural network is used to identify the control system (e.g. the input and output signals of the control system are used to train the network and map the input and output signals) and then used to develop a control strategy. The approach can be used for describing dynamical models, dynamic optimization and adaptive critic methods (e.g. reinforcement learning). Neural networks can be trained in many ways, such as with a supervised method where human expert experience is integrated into the network. This is one possible way of integrating expert knowledge about the bolt system and furthermore would allow error recognition capabilities.

2.3.3.1 Non-linear control using non-linear separation with neural network

Another research approach is based on neural networks where the non-linear components of the system are separated, as introduced in [46]. In this strategy the non-linear and linear components can be fed into a neural network which learns the system behaviour. The approach can be in particular useful to handle the non-linearity of the bolt system, however, uncertainties that were not present during the training process cannot be handled by a neural network approach.

2.3.4 Fuzzy control strategies implemented with linguistic rules using expert knowledge about the system without a numerical model

A control strategy based on a Mamdani fuzzy controller has been described in [47]. The Mamdani Fuzzy controller is based on membership functions and linguistic rules. This controller is in particular useful for the bolt tightening application, since it allows to integrate knowledge by using linguistic rules as well as allows to handle uncertainties using Gaussian membership functions.

Linguistic rules can be used to implement expert knowledge about the bolt tightening process into the control scheme. In general this will require an in depth analysis of the control system (e.g. the bolt system, the tightening tool as well as the transducer running the tool).

This is one of the most promising strategies for this application which will be analysed in this thesis. Using the expert knowledge on bolt tightening, the bolt tightening process can be split into stages; this control strategy allows integrating knowledge of each stage to complete the tightening process.

2.3.5 Fuzzy neural control strategies

A neural network can also be combined with a fuzzy controller. This can be done by using a feed forward neural network which approximates a fuzzy control algorithm.

In general a fuzzy controller can be implemented in a neural network using a local field. A local field is representing the linguistic rules and the weights can be a representation of the membership functions.

The neural network can be used to tune the fuzzy controller as well as extracting fuzzy rules from numerical data. Another possible use is a hybrid system where a neural network and a fuzzy controller interact in the control scheme.

Therefore, there are various levels on how a neural fuzzy controller can be implemented. The simplest way is using a neural network to match the input and outputs of a fuzzy controller (e.g. the surface of the fuzzy controller). Using this approach, the neural network may be trained on a numerical model of the control system. Furthermore, this approach can also be extended to an online learning algorithm, where the neural network is first trained on a numerical model and then further fine-tuned online on the physical control system. This allows in particular integrating approaches that can deal with uncertainties and non-linear behaviours of the control system which usually cannot be modelled using analytical methods.

2.3.6 Genetic algorithm combined with a fuzzy controller

A genetic algorithm can be combined with several other controller types, e.g. a fuzzy controller. The Genetic Algorithm can fine-tune the fuzzy controller and optimize its membership functions while the controller is running. This will constantly increase the control result to a certain defined threshold level.

In [48] a combination of a fuzzy controller with a genetic algorithm is introduced and used on a job-shop manufacturing system. The approach is used to automate the process of decision-making in manufacturing.

2.3.7 Control strategies using a PI controller combined with a Genetic algorithm

Another possible tightening strategy can be based on a linear controller, such as a PI controller, combined with a genetic algorithm to fine-tune the controller to the physical system, as described in [49].

Using a PI controller requires a pre-estimation of the PI gains. This can be done via trial and error or using a numerical model, such as a state space model. The pre-estimation can be then based on a Genetic algorithm and the numerical model. Once the PI gains have been estimated they can be implemented on the physical manufacturing system.

The performance may not be accurate enough and therefore the Genetic algorithm can be applied to the physical system as well, where it will learn the non-linearity of the manufacturing system and adapt to the changes and uncertainties, if present during this training period. This process is a continuous process and as soon as a different bolt system is introduced (e.g. when a different set of bolts is taken from the warehouse into the factory), the process will continue to adapt.

2.3.8 Control strategies implemented as a sliding mode controller

The bolt tightening system investigated in this thesis is nonlinear and therefore, control strategies which are purely linear may not perform very well on the system. Another strategy is to use a sliding mode control strategy [50].

For this control strategy, the torque and angle curvature will be analysed and piecewise linearized. In order to do this the curvature is subdivided into several parts where each parts non-linearity is low enough to be assumed to have linear behaviour.

Each of the subdivided parts can be then treated as an individual linear system, and linear control strategies can be implemented, such as PID.

Furthermore, the control concept can be set up with various different control strategies, such as fuzzy or state space control. This allows designing a precise controller for a non-linear control system. In particular, the non-linear parts of the bolt tightening system can be controlled using a suitable non-linear control strategy and the linear parts of the bolt system can be controlled using a linear controller.

Between each parts there will be a switching algorithm which switches between the controllers (also called sliding between controllers) depending on which part of the bolt system is currently controlled.

This control strategy may be very useful for the bolt tightening process, as it can be combined with an in depth analysis of the bolt tightening process.

1.3.9 Error Detection Capabilities for the tightening process

Another requirement for the bolt tightening process is error recognition. Error recognition needs to be divided into two parts, which is damage prevention due to errors and error detection.

Error prevention requires running the tightening tool on different speeds at different parts of the system. This allows avoiding mechanical damages at each part, for example, where a mechanical damage can be caused by a high tightening speed of the tool.

Error detection requires an actual feedback from the controller to the operator that there is an error event. With a sliding mode controller, this can be implemented using a fuzzy control strategy for each part of the bolt system or with another error detector which is implemented separately and runs on top of the controller to detect error scenarios.

Summarizing, a sliding mode control strategy may be a very suitable control strategy for the bolt tightening problem, as it clearly allows to address non-linearity and uncertainties based on an in-depth analysis of the bolt system.

2.4 Summary

This chapter reviewed manual tightening tools as well as control strategies and explored their suitability for the bolt tightening assembly process.

There are many different manual tightening tools, which are either mechanical ones with a spring system to set the desired torque as well as electronic tools which are based on a strain-gauge torque sensor which will disengage when the target torque is reached.

There are also automated industrial tightening tools, such as some with an integrated controller and industrial fieldbus interfaces as well as analogue tools which can be driven by an analogue input signal.

The chapter furthermore discussed and compared control concepts which can be implemented in a discrete control environment. It turned out that strategies based on fuzzy control can be in particular useful for the bolt tightening application providing the possibility to integrate knowledge of the bolt tightening process; furthermore the fuzzy-based control strategy can handle the non-linearity of a bolt tightening system.

A genetic algorithm based concept has also been shown to be suitable for the discussed problem; such an approach allows learning and adapting to changes in the physical bolt tightening process while the system in running.

CHAPTER 3

3. Chapter 3: Bolt tightening based on fuzzy control for wind turbine bearing assembly

3.1 Chapter outline

The modern wind turbine industry contains many core processes. One is the assembly of the bearings on wind turbine hub. The hub contains many bolts which connect the bearing for the blade to the wind turbine hub. The bolts ensure that the bearings are safely connected to the hub.

This chapter analyses the control problem of an automated bolt tightening process based on a fuzzy control strategy. For the control design, the tightening process has been analysed and divided into four stages which address the individual control requirements in each stage of the tightening process.

Moreover, errors need to be detected during the tightening process and the process needs to stop as soon as an error occurs to avoid any mechanical damage. This requires that the controller integrates additional knowledge about possible error scenarios related to each stage so that individual errors can be detected and classified.

Furthermore, the controller will be executed in real time on an industrial PC automation environment. This set up has also been used to validate the error capabilities of the controller.

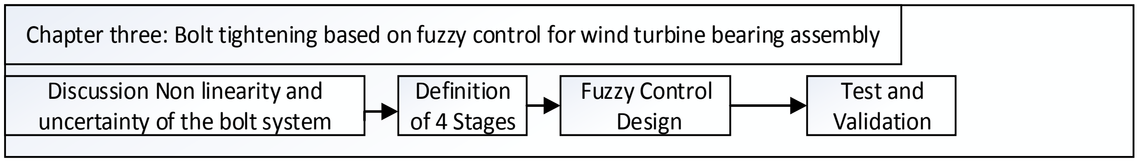


Figure 13: Chapter 3 structure

3.2 Bolt tightening strategies using fuzzy control

To handle the previously introduced complexity of the bolt system, a Mamdani-type fuzzy controller has been chosen, since it is model-free and allows the integration of expert knowledge using membership functions and linguistic rules into the control strategy [51] [29] [52] [53] [46] [42] [76] [40] [38] [78] [79] [80] [81] [82] [77] [83] [84] [85] [86] [87] [88]. The Mamdani fuzzy controller allows also implementing multiple inputs and outputs of output to actively control torque and angle values of the bolt-tightening tool. The linguistic rules combine the membership functions to provide an accurate desired output of the fuzzy controller.

To facilitate the design of the fuzzy logic controller (FLC), the process is divided into four stages according to mechanical properties. Knowledge of each stage is employed to come up with rule base and membership functions of the FLC. As an individual fuzzy controller is designed for each stage, nonlinearity can be clearly addressed and utilized for control design resulting in a well-performing system. To realize the fuzzy error detector for each stage, knowledge on potential error scenarios such as misalignment of nut on the bolt, mechanical damages of the bolt or the nut, incorrect thread types, sizes and so on, are defined in linguistic rules based on Mamdani FLC [16] [54] [29] [55] [56] [57]. Since different wind turbine hubs define different tightening specifications, a status determiner has been set which can set the parameters within the FLC to achieve the specified torque/angle [16]. The proposed FLC, error detector and status determiner are implemented on a real time industrial control system. Experiments are conducted to show the merits of the proposed control scheme.

3.3 Wind turbine components and suitable control methods

The wind turbine hub is made of three main components, which is the hub, the bearing and the pitch system. There are two types of pitch systems, hydraulic and electrical. The electrical pitch system is normally installed in the wind turbine hub [9].

The bearing is used to mount the wind turbine blades and to rotate them according to the wind speeds to different angles. The blades are rather heavy and due to variable wind speeds the bearings are installed using up to 128 bolt connections.

The general set up of the wind turbine hub assembly is shown in Figure 14.

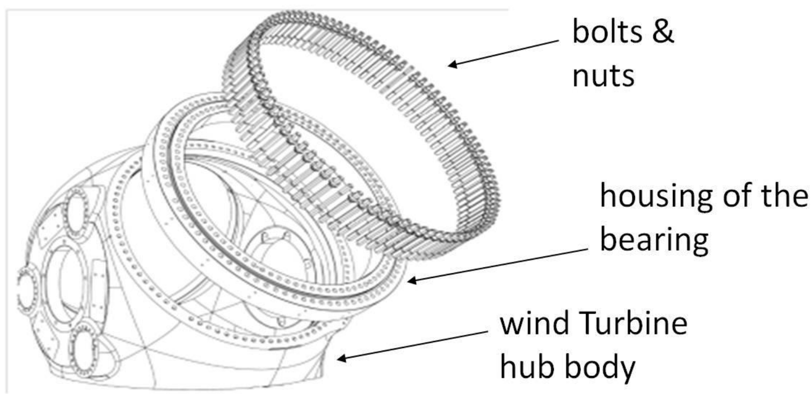


Figure 14: Overall assembly process (picture provided by Gamesa Corp.).

The picture above shows the three components and how they are all pre-assembled. The assembly is done in different steps, where the hub is placed in an assembly station, the bearing is placed on top of the hub and the bolts (or often referred to as stud bolts) are preinstalled by a robot arm.

3.3.1 Sequence of bolt tightening

The sequence of bolt tightening is essential for accurate bolt tightening as well as for assembly error detection. The process has been analysed and divided into four different stages where stage 1 targets the nut alignment on the bolt followed by partial engagement (stage 2), where the nut is rotated a few degrees until the threads of the bolt and nut engage with each other for a few degrees followed by full engagement (stage 3). In stage 3 the nut runs down until it touches the washer and the flange. The final stage (stage 4) is the actual tightening process there the nut runs to its final angle and final torque level. Once this is completed the final clamping force is applied to the flange.

The strategy is shown in Figure 15. The numbers in the figures are referring to the individual stages, e. g. stage 1, stage 2, stage 3 and stage 4.

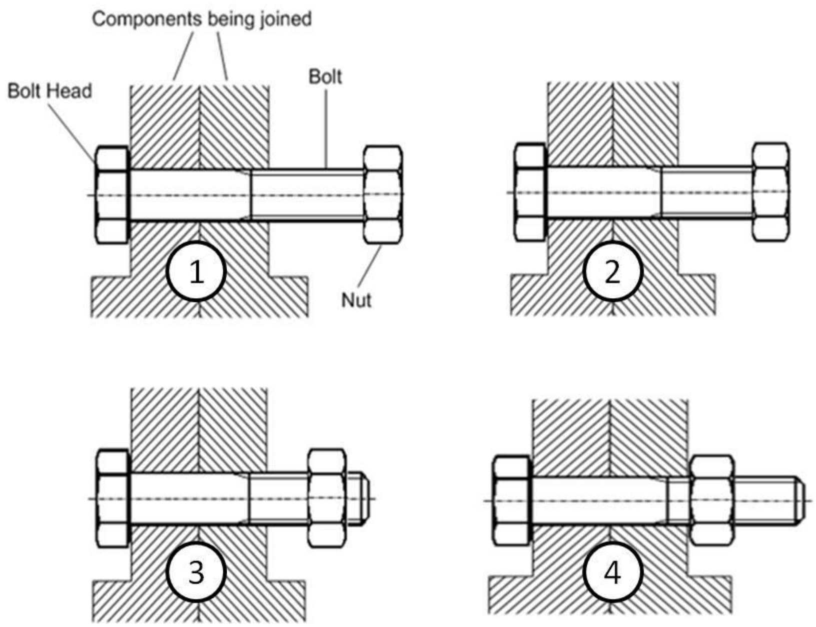


Figure 15: The 4 stages of the bolt tightening process [17]

3.3.1.1 Stage 1 bolt nut alignment

In this stage the nut is placed on top of the bolt using a robot arm. In the assembly process, the nut is picked by a robot arm using a magnet and then placed on top of the bolt. The nut runner is equipped with a magnet to grab the nut from a table where the nuts have been prepared. The general set up is shown in Figure 16.

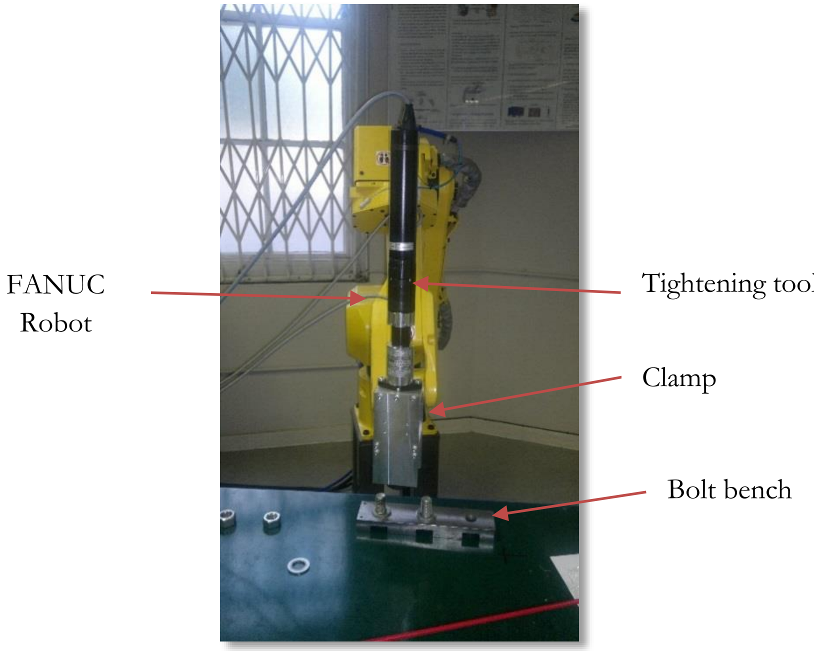


Figure 16: Robot set up

As it can be seen, the bolts have been preinstalled in the bolt bench so that the tightening tool can complete the tightening process. To overcome a backward torque problem, the tightening tool has been extended using a clamp system. When the tightening tool performs the tightening process, the torque is not only applied to the bolt itself, it is running through the whole robot arm and also works on the individual robot joints, where it may cause damage. To overcome this, the torque reaction system connects to the bolt bench and creates a closed system where the tool is locked onto the bolt bench. During the tightening the torque will stay in the bench/clamp system and no torque will be applied to the robot arm. This is shown in Figure 17.

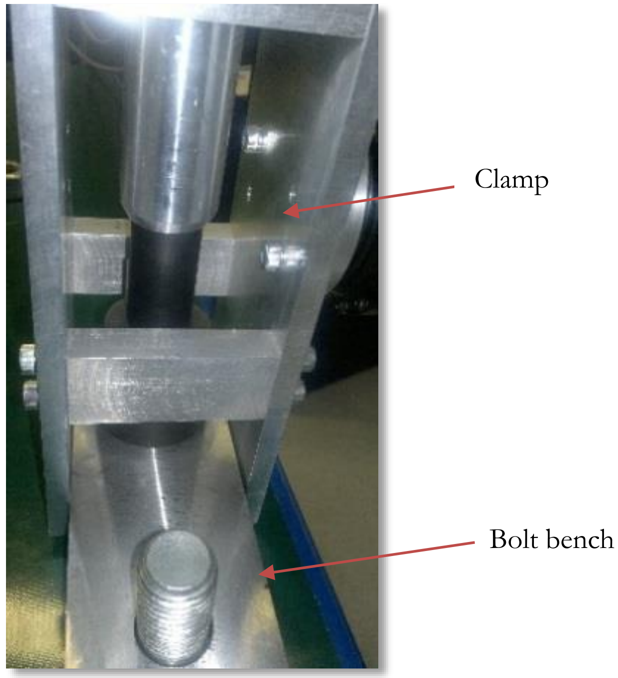


Figure 17: Clamp with bolt bench

The nuts are arranged in a row so that the robot can pick and place them, as shown in Figure 18. It shows the alignment for the experimental set up. In a factory environment the nuts are provided similar in a row and specific angle (so that the nut-runner can pick it).

Figure 18: Nut arrangement for the pick and place process

In general the nuts are aligned in a row and can be picked by the robot arm using a magnetic nut-socket. In this example, they are just placed on a table but feeding system can be employed which constantly provides bolts to the assembly system.

At the beginning of the tightening process the nut is meeting at the starting point, which is on top of the bolt. In Figure 15 the top left panel shows this process where the nut is placed on top of the bolt. It has to be ensured that the alignment angle is set up correctly and that the tightening process starts slowly to avoid any mechanical damages due to errors such as a misalignment of the nut or different types of bolts and nuts (e.g. metric and non-metric nuts).

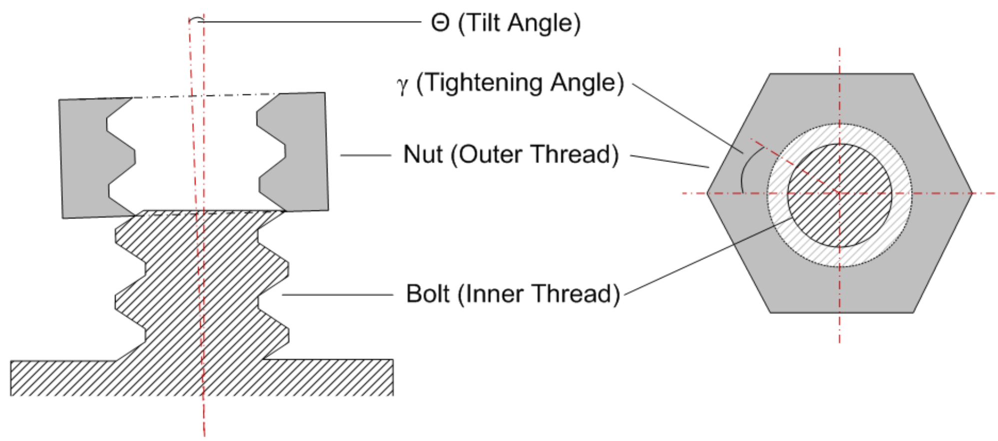


Figure 19: Alignment problems [47]

Figure 19 shows the alignment problem when the nut is placed on the bolt with an incorrect angle. This may be caused by the robot arm when the angle Θ of the place process is not correct or it may also be caused by an incorrectly installed bolt.

When the nut-runner is started to initiate the tightening process the nut may get jammed in this scenario. Therefore, the spinning speed is slow in this stage.

3.3.1.2 Stage 2 partial engagement

In this stage the nut is turned for a few degrees until the bolt and the nut thread meet each other. Stage 2 is shown in the second panel (top right) in Figure 15. In this stage it may happen that the nut gets jammed when the nut and the bolt have slightly different sizes or thread types, hence a few degrees of turning may be possible but then the nut will get jammed.

The described error scenario can be detected by measuring the torque level and compare it with the angular position. If the torque level is too high, it means that the nut is jammed and the tightening process has to be stopped immediately.

The misalignment problem (see Figure 19) is also a possible error scenario in this stage. It may happen that the nut can be turned for a few degrees even though it is not correctly aligned. In this situation the nut will get jammed on the bolt and may also get damaged.

Once the nut is tightened for a few degrees on the bolt, the stage is completed. The angular level is depending on the bolt size and is set according to the assembly specifications and may vary from the targeted assembly item to item.

3.3.1.3 Stage 3 - full engagement

At this stage, the nut is running down until reaching the flange and a maximum and steady friction level occurs (Figure 15, bottom left panel). Possible errors include cross threads on the bolts shaft and dirt between the threads which can be detected by unexpected higher torque. Monitoring the angular displacement of the nut is very important in this phase since it contains a feedback about how far the nut has travelled on the bolt shaft. Moreover this information supports the estimation of the effective bolt length, according to the assembly specifications and - based on the travelled distance of the nut – the detection of wrong or missing washer.

3.3.1.4 Stage 4 - final bolt tightening

The final tightening process starts as soon as the nut has reaches the flange. A further tightening will generate a clamping force between the flange and the nut (Figure 15, bottom right panel). The torque levels as well as the final angular position of the nut are provided within the assembly specifications. Accordingly, the consequent requirement of this stage is to apply certain values of torque within well-defined angular displacements and without exceeding the bolt tension limit. If the bolt exceeds the tension limit it will fail, therefore, this scenario causes an error.

3.3.2 Mamdani type control architecture

A Mamdani fuzzy controller was set-up, incorporating expert knowledge about the four stages of the bolt-nut tightening process through fuzzy rules. According to [21], the overall structure of the controller is:

$$\mathrm{MAMD}(x,y) = \bigvee_{i=1}^{n} (A_i(x) \& B_i(y) \tag{1}$$

where A_i and B_i are the concrete numbers or fuzzy numbers (e.g. low angle, desired angle) as a listing of n-possibilities. In the formula (1) the fuzzy numbers x can be seen as A_1 and y is B_1 <u>or</u> x is A_2 and y is B_2 and so on. Fuzzy rules can be integrated as a conjunction of implications:

$$\mathrm{RULES}(x,y) = \bigwedge_{i=1}^{n} (A_i(x) \rightarrow B_i(y) \tag{2}$$

In the Eq. (2) the rules have been set as a listing of n possibilities: if x is A_1 then y is B_1 and x is A_2 then y is B_2.

The application targets to be used in an industrial environment. Therefore the proposed overall architecture uses a Programmable Logic Controller (PLC) [58] [59] system which integrates MATLAB/Simulink programming language (by the Mathworks Inc.) on a real time Beckhoff TwinCAT 3 software automation system. The PLC is connected to an industrial robot arm (the Fanuc M6iB model), which is equipped on the end-effector flange with a tightening tool (model DSM BL 57/140 MDW). For testing the tightening process, a bolt bench with 3 bolts is set up, as described in Appendix A. The inputs and outputs of the fuzzy controller are the tightening tool angular position (measured by means of an integrated encoder) and the torque (measured using an integrated strain gauge sensor with a current output to measure the torque) respectively, as well as an error input used for tension limit detection. The fuzzy controller output is a voltage signal, in the ± 10 V range, which sets the spinning speed of the tool. A supervision signal is also defined to identify the errors.

The number of each fuzzy controller rules and membership functions for each stage may vary depending on the stage requirements. Based on the current stage, the error can be promptly classified, the assembly process can be stopped and the potential damages to the assembly item can be avoided. Furthermore the fuzzy controller runs the tool on a lower speed during the most critical phases in order to prevent physical damages. Compared to an over-layered quality control system this approach has the advantage that the tightening tool can respond faster to an error scenario. Compared to a PID control strategy, the tool runs on maximum speed when the tightening process is started (as at this point there is the maximum control error), therefore, in an error event, it may not stop adequately and cause damage to the bolt system. This is why it is necessary to develop a control strategy which integrates the quality control requirement automatically. The final control target is to reach a specific clamping force which cannot be actively controlled as there is no real time information about the clamping force available. The torque angle tightening technique allows reaching a specific clamping force by defining a tightening angle as well as a tightening torque [16].

By using a TwinCAT 3 system, the fuzzy controller is cyclically executed and sends the results to the PLC system. In fact, the PLC is connected to the corresponding tightening tool where the overall controller architecture is shown in Figure 20 and Figure 21.

The control signals are sent back to the PLC in real time and the latter drives the tightening tool. It is important to notice that this approach has the advantage that (a) different fuzzy controllers can be chosen by the PLC to be used for different bolt types and (b) several tightening tools can be integrated by calling the fuzzy controller several times.

Membership functions can be set by defining the required control values by the PLC when the fuzzy controller is called. Therefore, the error can be minimized by using the membership functions (which define the targeted control values) and linguistic rules. The generic control loop is illustrated in Figure 21. It needs to be considered that the error (i.e. the difference between the real torque/angle and their desired values) is estimated internally within the fuzzy control block by means of the defined membership functions and linguistic rules; therefore, there is no additional error feedback into the controller.

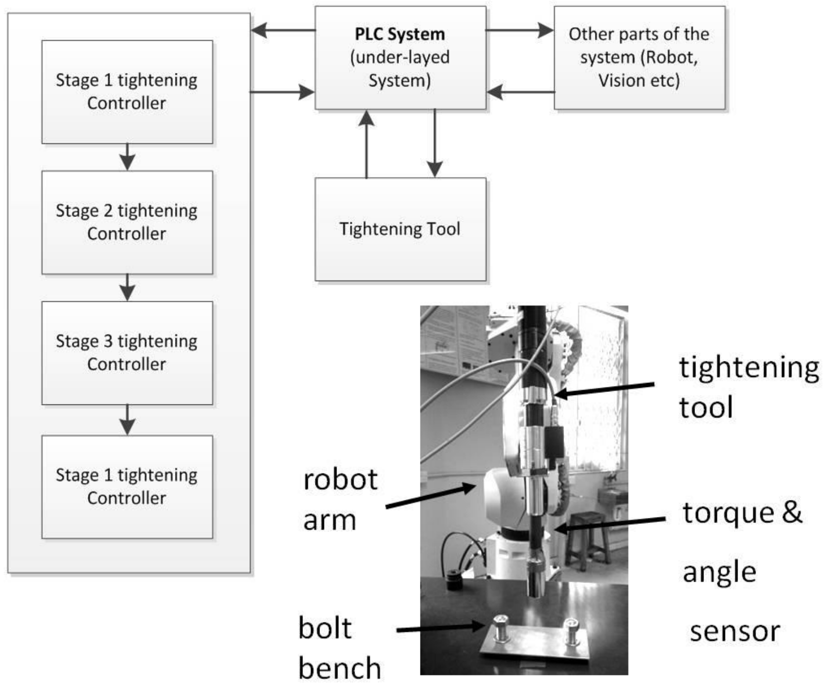


Figure 20: Overall controller architecture

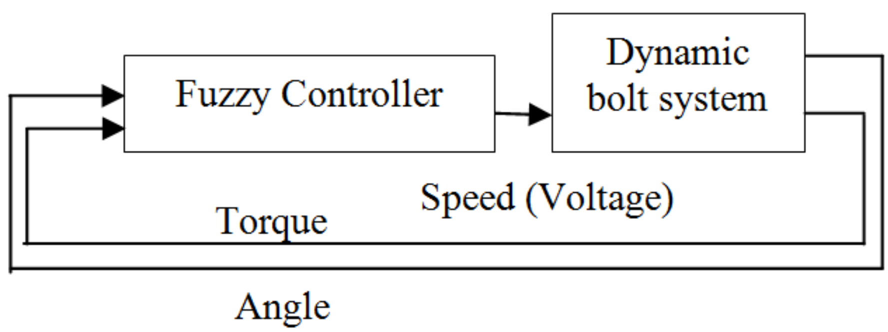


Figure 21: Generic control diagram stages 1-4.

3.3.2.1 Stage 1 control strategy (bolt/nut alignment)

A MIMO fuzzy controller with two inputs (torque and angle as sensing inputs) and two outputs (voltage for setting the tool's speed and error for returning the control voltage for the tool) has been designed.

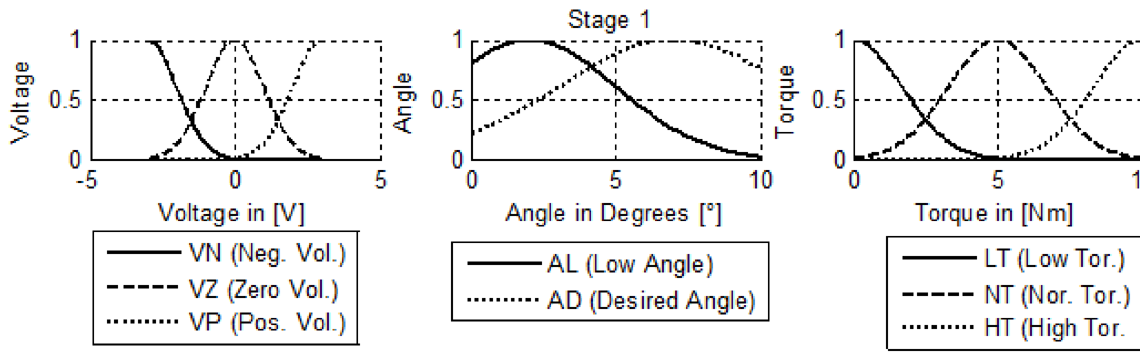


Figure 22: Membership functions of stage 1

In stage 1, the input torque of the fuzzy controller contains three Gaussian membership functions named "low torque (LT)", "normal torque (NT)" and "high torque (HT)" conditions; the input angle contains two membership functions, which are called "low angle (AL)" and "desired angle (AD)"; the output voltage contains three membership functions, namely "negative voltage (VN)", "zero voltage (VZ)" and "positive voltage (VP)".

All membership functions are in the Gaussian shape as shown in Figure 22. The fuzzy rule set is reported in the Table I, where the fourth column refers to the output of the fuzzy error detector, which generates either "true (T)" status, indicating erroneous condition, and "false (F)" status, indicating proper operation. In the first case, the fuzzy controller switches off the output voltage and reports an error output by sending a supervision signal to the PLC. Operatively, the tightening tool rotates until it reaches the starting position (where the bolt and the nut thread meet); then the torque slightly increases and the control target is satisfied.

3.3.2.2 Stage 2 control strategy (partial engagement)

The stage 2 fuzzy controller has a structure similar to the previous one, with the membership functions of the angle which is adapted to the desired angular range (Figure 23) in such a way that if a high torque scenario arises, then the voltage output is set to zero and an error output is returned. The membership functions are linked using the same linguistic rules as reported in stage 1 and Table 1. This stage is entirely 'angle based', since only 3-5 entire turns of the nut are required for the stage completion.

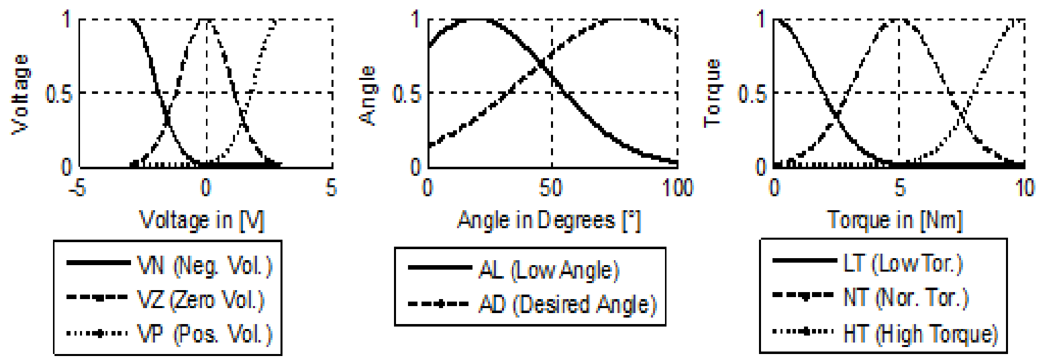


Figure 23: Membership functions of stage 2

The abbreviations in Table 1 have the same definition as defined in Figure 23. The error detector sets the error flag, which can be either 'true' (T) in an error event or 'false' (F) if no error occurred.

INPUTS		OUTPUTS	
Angle	Torque	Voltage	Error
AL	LT	VP	F
AL	NT	VP	F
AL	HT	VZ	T
AD	LT	VZ	F
AD	NT	VZ	F
AD	HT	VZ	T

Table 1: Linguistic rules stage 1

3.3.2.3 Stage 3 control strategy (full engagement)

In stage 3, the fuzzy controller contains also 2 inputs (torque and angle for sensing) as well as 2 outputs (the voltage and error signal for actuation). Compared to stage 1, the angle range has to be redefined to cover the expected run down of the bolt's shaft; moreover the errors have to identify any possible high torque scenarios, which may be caused by cross thread on the shaft. Accordingly, the membership functions have been set as reported in Figure 24.

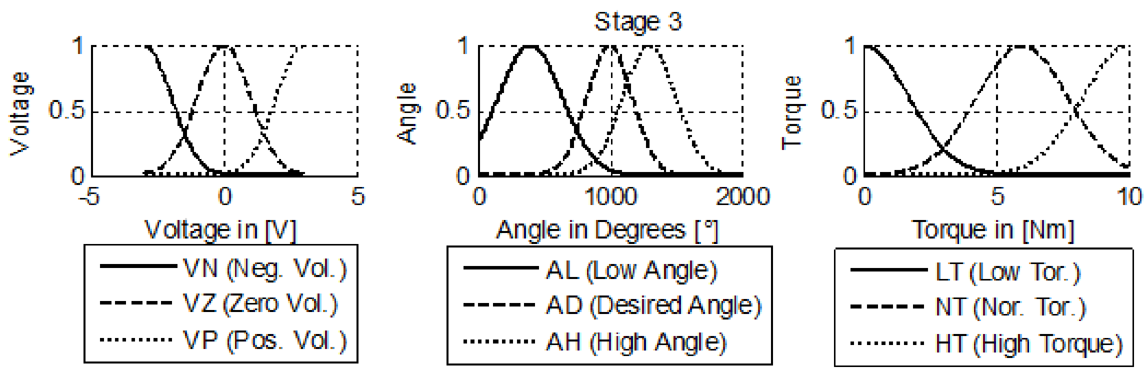


Figure 24: Membership functions of stage 3

INPUTS		OUTPUTS	
Angle	Torque	Voltage	Error: F=false T=true
AL	LT	VP	F
AL	NT	VP	F
AL	HT	VZ	T
AD	LT	VZ	F
AD	NT	VZ	F
AD	HT	VZ	T
AH	LT	VZ	T
AH	NT	VZ	T
AH	HT	VZ	T

Table 2: Linguistic rules stage 2 and 3

The abbreviations in table 2 are defined in Figure 24.

Due to the presence of the friction between the bolt and nut threads, the baseline of the torque value within the fuzzy rules has to be slightly increased (as the nut's thread is now fully set on the bolt's thread) and furthermore the angle region has to be re-defined to estimate whether a correct washer has been installed (a missing or false washer would cause a high angle scenario). A high torque scenario within the low angle region would be indicative of a problem - as the situation of a cross thread on the bolt or a too short installed bolt - and must stop the tightening action. Due to all these concerns, more membership functions and linguistic rules have to be defined within this stage, as it is reported within Figure 24.

3.3.2.4 Stage 4 control strategy (tightening process)

The fuzzy controller in stage 4 tightens the nut to the final desired torque and within a specified and desired angular range. Here the tension limit has to be preserved (meaning that the bolt cannot be over tightened due to a wrong bolt installation). Therefore, the controller is set up by using three inputs (torque, tension-limit and angle for sensing) and two outputs (voltage to set the tools speed and one supervision signal for the error and tension-limit detection). Both outputs are based on the Gaussian functions and the error occurrence is detected by combining both the outputs. Two comparators are linked to the tightening and tension limit outputs, respectively: this set up enables detection of errors and tension limits.

Three membership functions are assigned to each of the inputs. The error recognition should detect if the bolt reaches its tension limit by deviation of the torque: as soon as the torque velocity remains constant and the angle is still increasing, then the limit value has been reached and the process must stop, either with an error (if the torque has not been reached) or with no error (if the torque has been reached and the angular position is within the desired range).

Furthermore, in this stage, the fuzzy controller returns to the PLC system whether the process has been successfully completed or not. According to these observations, the membership functions are both Booleans and they are implemented in addition to the membership functions introduced in Figure 25. "Reached tension limit (RT)", "close tension limit (CL)" and "below tension limit (BE)" functions are also introduced.

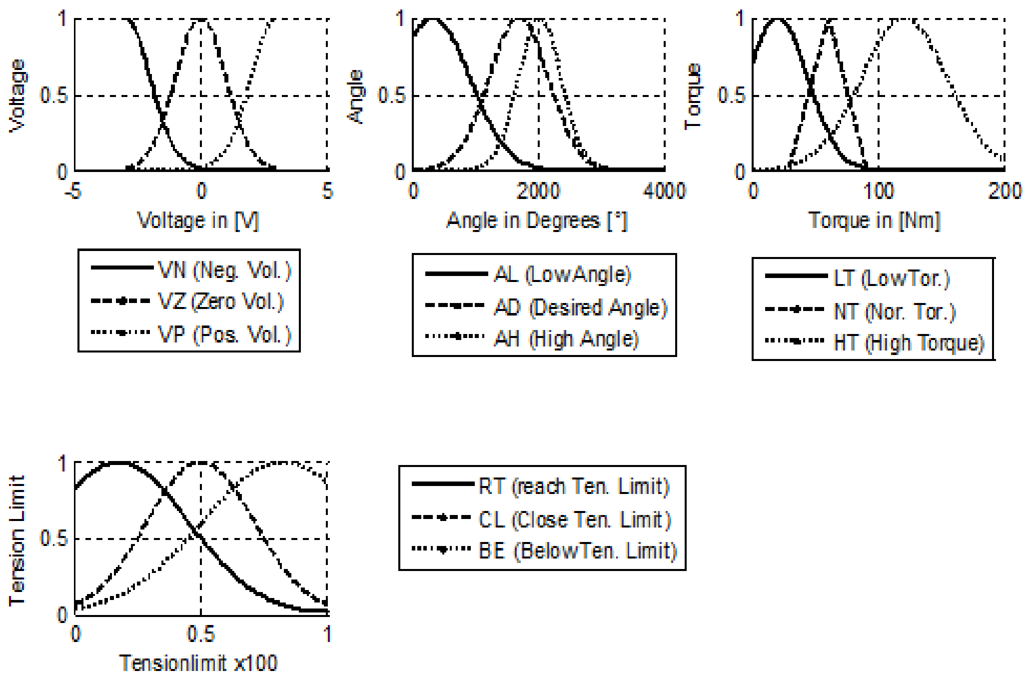


Figure 25: Stage 4 membership functions

The Boolean outputs of the tightening (TIGH, showing that the tightening process is active if true) and of the tension limit (TL, showing tension limit reached if true) are supervision signals set by the fuzzy controller; then a set of 27 linguistic rules has been set up to cover the required actions (Table 3). Even though the membership functions are Gaussian, the output is Boolean for the tension limit. The overlaid system receives a Boolean which is either true in an error scenario or false if the process has been completed without errors. The Boolean conversion is done by using a comparator which becomes true as soon as the membership value rises up to 1.

The combination of all membership functions and linguistic rules leads to the overall fuzzy controller shape as reported in Figure 25.

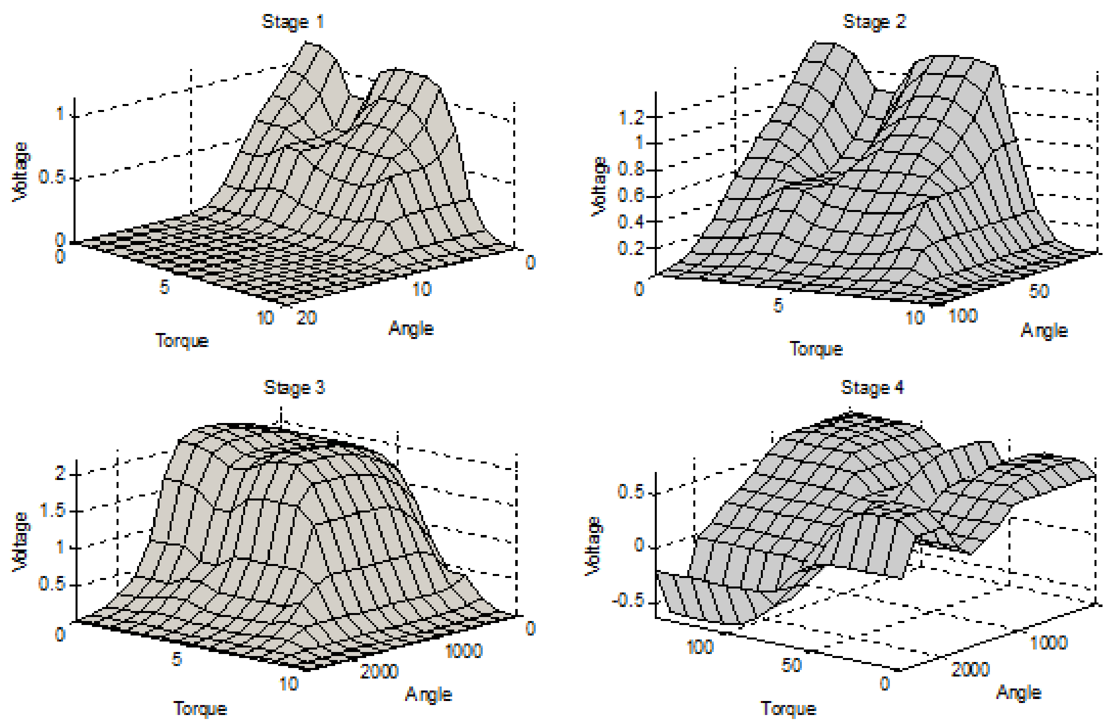


Figure 26: From top left to bottom right panel, the Stage 1, 2, 3 and 4 membership functions with their linguistic rules, respectively

Table 3 shows the linguistic rules for stage 4. The abbreviations are defined as in Figure 25, 'TIGH' is the Boolean showing the tightening is completed (true 'T' if the process is still running, 'F' if the process is finished) and 'TL' describes if the tension limit of the bolt has been reached (true 'T' for reached, false 'F' if not). The normal torque (NT) also describes the control target, e.g. 60Nm.

INPUTS			OUTPUTS		
Torque	Angle	Tension Limit	Voltage	TIGH	TL
LT	AL	BE	VP	T	BE
LT	AD	BE	VP	T	BE
LT	AH	BE	VP	T	BE
NT	AL	BE	VP	T	BE
NT	AD	BE	VZ	F	BE
NT	AH	BE	VZ	F	BE
HT	AL	BE	VP	T	BE
HT	AD	BE	VN	T	BE
HT	AH	BE	VN	T	BE
LT	AL	CL	VP	T	CL
LT	AD	CL	VP	T	CL
LT	AH	CL	VZ	F	CL
NT	AL	CL	VP	T	CL
NT	AD	CL	VZ	F	CL
NT	AH	CL	VZ	F	CL
HT	AL	CL	VP	T	CL
HT	AD	CL	VZ	F	CL
HT	AH	CL	VZ	T	CL
LT	AL	RT	VZ	F	RT
LT	AD	RT	VZ	F	RT
LT	AH	RT	VZ	F	RT
NT	AL	RT	VZ	F	RT
NT	AD	RT	VZ	F	RT
NT	AH	RT	VZ	F	RT
HT	AL	RT	VZ	F	RT
HT	AD	RT	VZ	F	RT
HT	AH	RT	VZ	F	RT

Table 3: Stage 4 linguistic rules

3.4 Validation

3.4.1 Experimental set-up

The previous section introduced a 4-stage fuzzy controller performing bolt tightening with error detection. In order to validate the system in an industrial software and hardware environment, the controller was initially implemented by using the MATLAB/Simulink Programming Language and then imported into the Beckhoff TwinCAT 3 system by means of MATLAB coder. The controller is then executed at a cycle frequency of 2 KHz (i.e. with a cycle time of 500μs). This cycle time was selected due to the speed requirement of the tightening process.

The tightening tool (model DSM BL 57 – maximum torque performance of 140 Nm) was mounted to the end-effector of the Fanuc M6i-B robot arm. During a regular tightening procedure, the robot picks and places an M24 nut on the top of another M24 bolt; rotational tightening speed was realised by the voltage command, whereas integrated optical encoder and torque sensor - integrated within the tool - measured the angle and torque, respectively. Finally, the fuzzy controller was interfaced with the angle and torque values, as the inputs, and with the voltage control and error code as the outputs.

A washer sensor (MecSense KMR 50 KN), for measuring the clamping force, was also inserted between the nut and the flange in order to measure the effective performance of the tightening process. Generally, the clamping force depends on multiple factors such as the applied torque, the relative angular positions between the bolt and nut threads, the geometric and mechanical characteristics of their contact surfaces [60] [29]. Usually this sensor is not installed in the physical assembly line and is only used for verification.

3.4.2 Validation Scenarios

Regular tightening processes as well as sessions for testing the error detection capabilities of the control algorithm were performed. In particular, to test the error detection capabilities diverse error scenarios were set up during the tightening processes. The error feedback is set up by using a Boolean flag within the PLC which returns the actual torque, angle and stage values as soon as an error detection occurs.

Finally, the performance of the fuzzy controller was also compared with the ones of a classical industrial PID controller for normal tightening process.

3.4.3 Error recognition performance

To validate the controller and its capability in order to detect the errors, six experimental sessions were performed within different Scenarios (S1 – S6). For each session, at least eight trials were performed for each session. At the beginning of each trial, the tightening tool loaded the nut and was positioned in front of the bolt; then the controller was started and runs until the tightening was completed or until any error detection occurred.

The desired torque level is depending on the application's specification. In the wind turbine assembly high torques are required for the hub assembly based on the specifications. Based on the specifications, the PLC sets the membership function parameters for the desired torque and angle and starts the controller.

The six scenarios (S1-6) are intended to replicate the typical errors which may occur while wind turbine assembly is performed by an operator. These scenarios were conceived and designed in order to cover diverse corresponding error detections within the four stages of the tightening process. These are the six scenarios that were experimentally validated:

1. Regular tightening (S1): no error detection was expected within this scenario, since a correct M24 nut was positioned within the tightening tool and in front of an M24 bolt.

2. Misalignment error (S2): the tool and the nut were erroneously positioned with respect to the bolt in order to replicate the misalignment (Figure 27); the error detection was expected to occur at Stage 1.

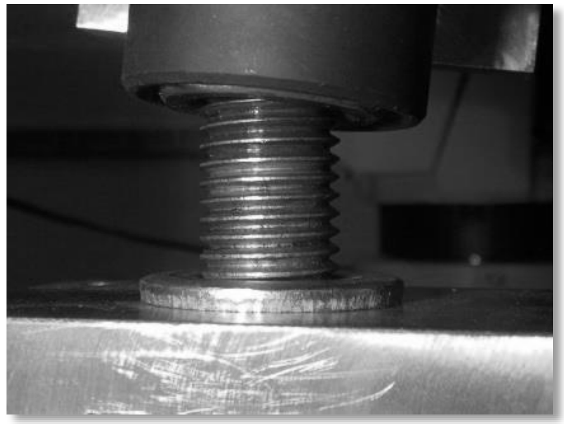


Figure 27: Misalignment error: The robot places the nut on a faulty angle which causes the nut to be stuck on the bolt as soon as tightening process is started.

3. Jamming error (S3): a non-metric nut was tightened on an M24 bolt, therefore an error was expected to be detected at Stage 2, since the torque level would rise up to an undesirable level at this stage. The threads of the nut and the bolt were also expected not to grab on to each other, due to their different geometric shapes.

4. Insertion of two washers (S4): in this scenario one additional washer was included to the original washer of the bolt (Figure 28) and the system was expected to recognize its presence during Stage 3 or early in Stage 4 because the torque level would reach a too high value within Stage 3. The torque level would also stay high with a low angle level in Stage 4.

Figure 28: The 'two washers' error scenario: The 2nd washer will cause an error detection within Stage 3 or 4, since the desired angular position of the nut will not be reached.

Figure 28 shows a "two washers' error scenario. In this case the tightening angle cannot be reached according to the assembly specifications. Therefore the torque level increases before a specified angle is reached and an error is detected.

5. Missing nut (S5): to simulate a mistake of the operator, the nut was removed from the tightening tool (Figure 29). In this situation the controller error detection was foreseen at stage 3, because no increase in the torque was expected and the angle should run in the high angle range of stage 3. As shown in Figure 29, the tightening tool spins on the bolt as there is no nut which can cause an increase of the torque value. This should cause an error since the controller is expecting a rise in the torque during the stage 3. Finally, the nut runner is touching and spinning on the washer since there is no nut in the set-up.

6. Wrong bolt versus nut (S6): the proper M24 nut was replaced with an inappropriate M14 nut; therefore a smaller nut was used with respect to the M24 bolt. In this situation, the controller runs into stage 3, as the torque level remains on a low level and the 'wrong bolt' error should finally occur at stage 3. This type of error can also imply that a too small nut-runner was installed; hence, the nut needs to be picked and placed on the bolt.

Figure 29: Missing nut scenario: the tool is touching the washer as soon as it is placed on the nut, because of the missing nut.

During all the experimental tests, the following two parameters were used to measure the system performance:

- Percentage of successful trials, i.e. trials in which at least one error message was detected.
- Percentage of trials in which the error message occurred within the proper and expected stage.

3.5 Results

3.5.1 Scenario 1 - regular tightening

In this experiment, the control target is to reach a final torque value of 60 Nm as well as a tightening angle of approximately 2000°. This angle may change according to the installation of the bolt which causes different starting angles, depending on how the operator positions the bolt in the hole.

Figure 30 summarizes the typical time history of the angle, torque, clamping force, stage and error during the trials within the regular tightening scenario (S1): the five black stars report the stage transitions, namely the beginning of the trial, the end of 1^{st} to 4^{th} stage and the trial end (1^{st}, 2^{nd}-5^{th} and last markers, respectively); in the bottom right panel of the figure, the stage and error are shown in continuous and dashed grey lines, respectively: as it is shown in the figure, as soon as the fuzzy controller encounters any forewarning conditions, the error flag changes the status from 0 (i.e. no error detection) to 1 (error detected).

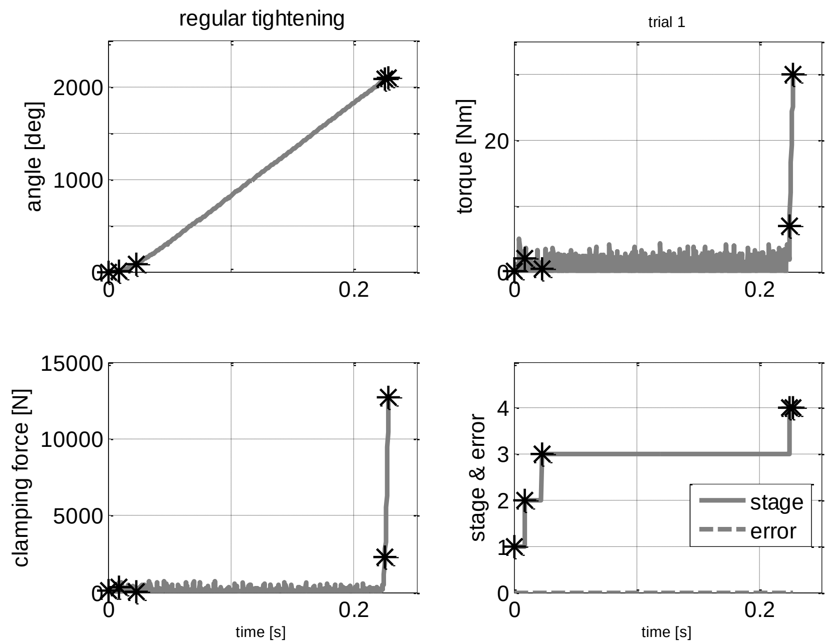


Figure 30: One trial (out of 8) of regular tightening (S1): from top left to bottom right panel, the angle, torque, clamping force, stage and error time patterns, respectively. Black stars report the stage transitions.

In summary:

- The complete regular tightening process took less than 0.5 s to be completed within the 4 stages.
- The process is always completed with no error detection.
- At the end of Stage 4, the average value of the tightening torque is tuned around the target value of 60 Nm (bottom right panel, Figure 31), whereas the angular position is largely distributed around 2000° (top right panel, Figure 31), due to variations in the initial installation of the bolt.
- The order of magnitude of the clamping force is more than 13.5 KN (bottom left panel of Figure 31). This is the average targeted clamping force based on the torque/angle tightening algorithm, based on the targeted torque and angle levels.
- The stage-by-stage time transition is quite regularly distributed on stage 1 and stage 2, whereas it is more extended on the stage 2 and 3 (top left panel, Figure 31). This is due to the run-down phase of the nut when it is driven down to the flange.

To quantify these observations, the mean and two times standard deviations values of time, angle, torque and clamping force were calculated at each stage transition and presented in Figure 31 and Table 4.

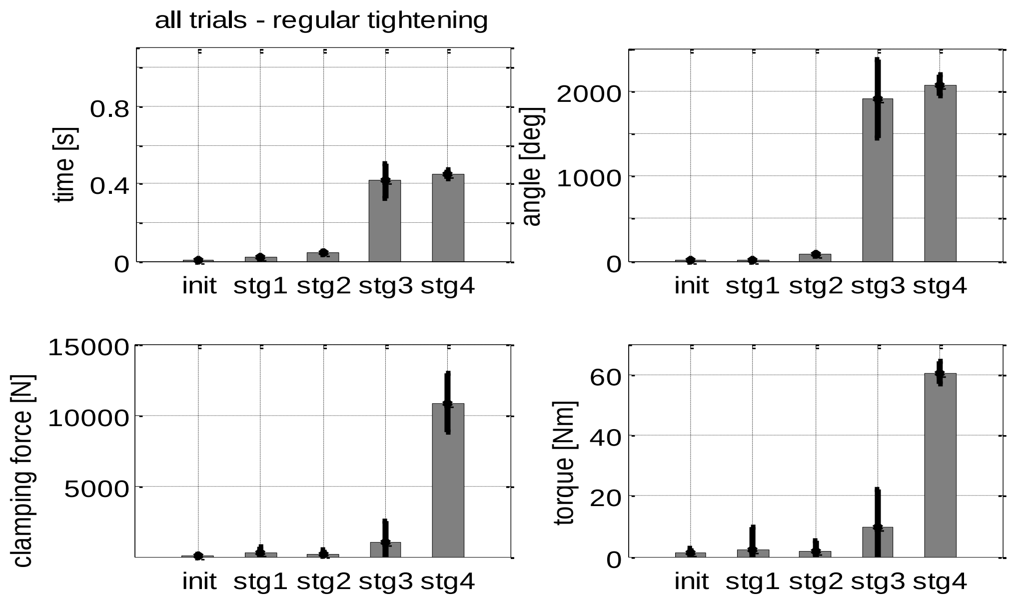


Figure 31: Average distribution (in grey colour bars) and two times standard deviations (black lines) of regular tightening (Scenario 1): from top left to bottom right panel, the time, angle, clamping force and torque respectively.

The clamping force has been measured using a physical clamping force sensor.

	INIT	END OF STAGE 1	END OF STAGE 2	END OF STAGE 3	END OF STAGE 4
Time [s]	0	0.018 ± 0.002	0.046 ± 0.002	0.414 ± 0.089	0.446 ± 0.024
Angle [°]	0.111 ±0.67	8.000 ± 0	82.444 ± 1.76	1908.000 ± 455.5	2066.000 ± 119.15
Clamping force [N]	69.338 ± 76.360	245.335 ± 450.532	178.753 ± 335.286	994.063 ± 1543.632	10853.478 ± 2120.51
Torque [Nm]	1.238 ± 1.41	2.094 ± 7.67	1.973 ± 3.45	9.765 ± 12.16	60.6 ± 3.51

Table 4: Regular tightening: mean and two times standard deviation of time, angle, clamping force and torque at each stage transition

- These results reflect and match the effective targets of the membership functions of the fuzzy controller. More specifically, the averaged time at which each stage transition occurs is equal to 0.018, 0.046, 0.414 and 0.446 s at the end of stages 1, 2, 3 and 4, respectively (Table IV); the two times standard deviation is always less than 5.4 % of the average, except in the beginning of Stage 2 and 4 (11.2 % and 21.5 %, respectively).
- At the end of the stage 1, the distribution of the angular position is quite large, because of the trial by trial differences of the initial mechanical alignment between the tightening tool and the nut with respect to the bolt; no variability of the angle is found at the beginning of the next stage (8 ± 0°) and during the other transitions the percentage of angle variation reduces at less than 6 %, except in the beginning of stage 4 where it is reduced by 23.9 %.
- The clamping force is the result of the combination of multiple nonlinear factors and therefore it is quite hard to be predicted. Nevertheless, a significantly low distribution of the clamping force is registered at the end of the tightening (19.5 %), meaning that - thanks to the fuzzy controller - the process is highly repeatable, i.e. the fuzzy controller is effective in

dealing with uncertainties. This latter result is a clear sign of the system capability in order to simultaneously reach the desired tightening force at the desired angular position of the nut with respect to the bolt and flange, with an error distribution which stay at values of 6 % and 5.8 %, respectively.

These fuzzy controller results were compared to the ones produced by a PID controller, where the proportional, derivative and integral gains were obtained by a trial and error process for good performance. The PID controller was employed for all 4 stages during 8 trials of regular tightening. The average results of both the fuzzy and PID controllers are plotted into a Gaussian distribution of the final torque and angle, as shown in Figure 32. The results may change for the PID controller hence the gains may be modified if the bolt system changes.

As it can be seen, within the Gaussian distributions, the accuracy of the fuzzy controller on the desired torque level is higher than the one of the PID controller. In fact, the mean $\pm$ standard deviation of the torque and angle of the fuzzy controller are equal to 60.553 ± 3.507 Nm and $2066° \pm 119.147°$, respectively, whereas the same parameters of the PID controller are equal to 61.10 ± 5.25 Nm and $2100° \pm 184°$, respectively. As mentioned before, this is caused by the uncertainty of the angle which slightly varies due to the installation of the bolt. The fuzzy controller can address this issue by using expert knowledge incorporated in the rule base and membership functions. The final value is within a tolerance band and may be further improved if more rules and membership functions are introduced. Nevertheless, this approach may make the design of the fuzzy controller more sophisticated and therefore increases its computational cost.

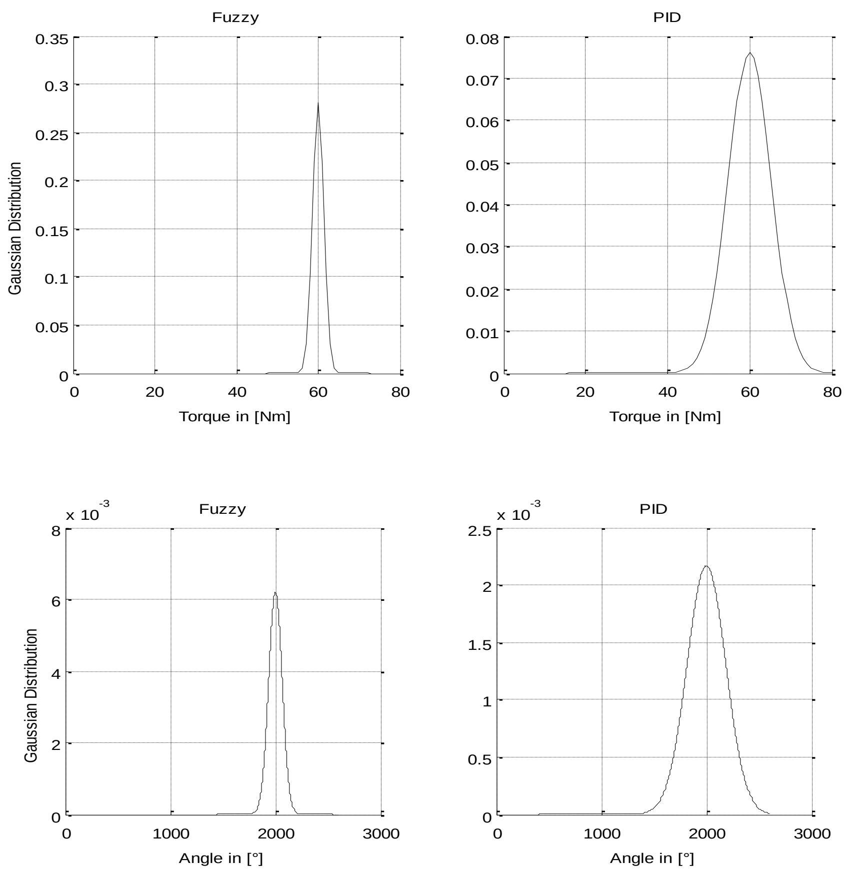


Figure 32: Comparison of the fuzzy controller to the PID controller in terms of final torque of regular tightening during 8 trials (left and right panels respectively).

Figure 32 shows that the confidence level is higher and the reliability for the fuzzy controller is better.

Furthermore, five more experiments have been completed using 70 Nm and 2100° as target values. Indirectly, the control target is the clamping force, which normally cannot be measured in real time during the tightening process hence there is no sensor washer in the assembly line to measure the clamping force.

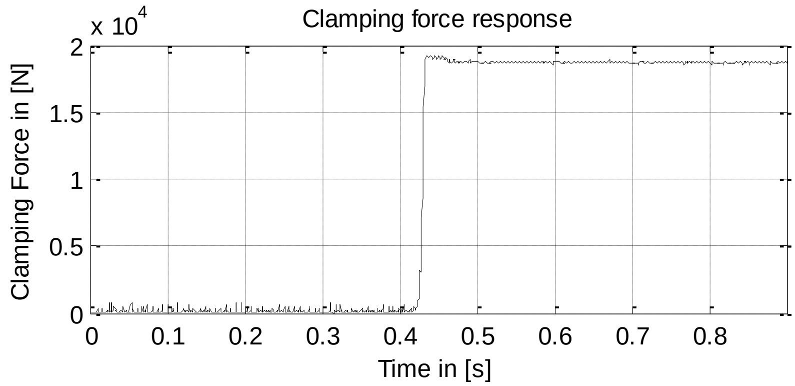


Figure 33: Final Result for the clamping force

Figure 33 shows the resulting clamping force after completion of the tightening process. The time delay is caused as the nut runs down from stage 1 to stage 4. The times may differ from the previous experiments since two new target values have been selected.

During the tightening process the more torque is applied the more the bolt is twisted [61]. As soon as the tightening process is completed, the material relaxes, which means that the nut moves slightly back from its position (until it gets stopped by the friction between the flange, washer and the nut).

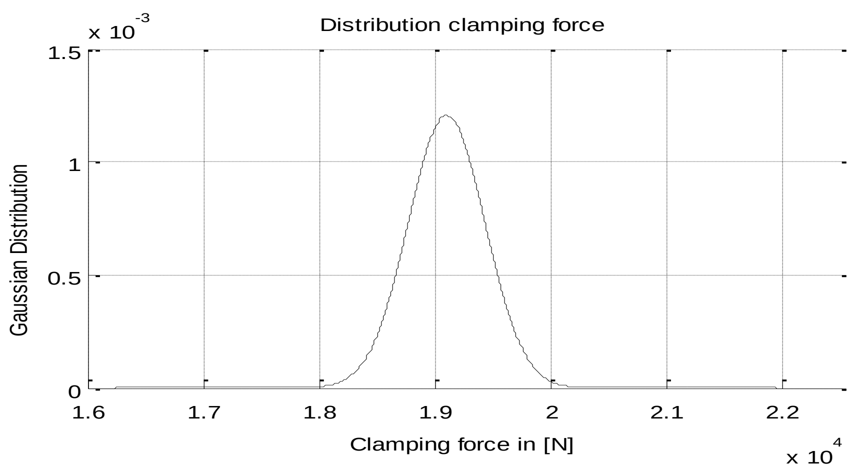


Figure 34: Gaussian distribution for 70 Nm, 2100°

ERROR SCENARIOS	TIME [S]	ANGLE [°]	CLAMPING FORCE [N]	TORQUE [Nm]
S2: Misalignment	0.005 ±0.007	0.7 ± 0.966	-	12.143 ± 4.842
S3: Jamming	0.016 ±0.016	7.2 ± 5.147	-	2.032 ± 5.098
S4: 2 washers	0.331 ±0.010	1481 ±41.070	1618.32 ±2243.610	12.88 ±16.060
S5: Missing M24 nut	0.484 ±0.055	2101.3 ±2.119	156.107 ±484.998	0.043 ±3.385
S6: Big nut	0.481 ±0.071	2081.60 129.042	92.748± 349.256	0.952± 4.255

Table 5: Average and two times standard deviation of time, angle, clamping force and torque as well as error detection

Figure 34 shows the Gaussian distribution for five experiments at the end of the settling effect. It can be seen that the clamping force can be reached on a reliable level, even though it cannot be controlled in real time by using the torque/angle tightening technique [15].

3.5.2 Scenario 2 – misalignment error

During all the trials of S2, the controller properly detected 100% of misalignments statuses within the proper stage, namely the 1st one (Table 5). At the error event, the average and two times standard deviations of the angle, and torque were registered, while no clamping force was detected because of the expected and early stop of the tool at Stage 1. All detections were discovered within the first 0.01 s (namely, 0.005 ± 0.007 s) of the tightening process, with the tool rotated of less than 1° and an applied torque of 12.1 ± 4.8 Nm.

ERROR SCENARIOS	DETECTION [%]	DETECTION STAGE [%]			
		Stage 1	Stage 2	Stage 3	Stage 4
S2: Misalignment	100	100	-	-	-
S3: Jamming	100	10	90	-	-
S4: 2 washers	100	-	-	33	67
S5: Missing M24 nut	100	-	-	100	-
S6: Small bolt, big nut	100	-	-	100	-

Table 6: Percentage of successful detection over all the trials within each stage where the error was expected for each scenario

Table 6 describes the error detection distribution in [%] over each stage. It can be seen that the error has been detected for each scenario.

3.5.3 Scenario 4 – insertion of 2 washers

In this scenario, two washers were placed as shown in Figure 28 and the tightening process was started. The controller exhibited a 100 % performance over all the nine trials. The error was detected within the expected stage (i.e. stage 3), in three out of nine trials (33 % of performance). In all the other six trials the error it was detected at the beginning of the Stage 4. Because during seven trials the nut and washers touched the flange, the variability of the clamping force was a little bit spread out (139%), whereas transition time and angle were centred on their averaged values (3 %) and deviation of the torque reached the value of 125

%. Similarly to Figure 31, a representative figure of the time history of all the parameters during the 'insertion two washers' scenario' is reported in Figure 35.

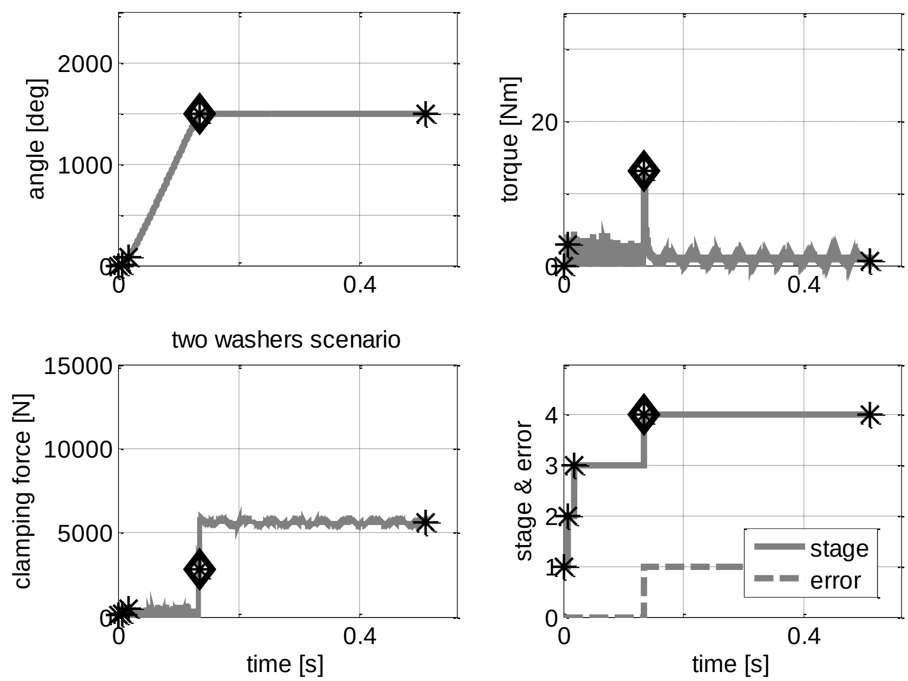


Figure 35: The two washers' error detection occurring at Stage 4 during one trial (out of 9).

3.5.4 Scenario 5 – missing nut

The 'missing nut' was found in all the trials (100 % of performance) and within the expected stage, namely the 3rd one (100 % of performance – Table 5). Table 6 reports the time history of the average values of the angle, clamping force, torque and error during one representative trial. Consistent with the membership functions of the fuzzy controller, a very low variability of the angle was found (0.1 %), whereas the distribution of the torque and, as a consequence of the clamping force, was rather high. However, since the final torque and the angle levels are distorted and altered during an error scenario, these values can be disregarded.

3.5.5 Scenario 6 – wrong bolt vs. nut

Within this latter scenario, a 100 % of performance was registered both in terms of the detection of the error within all trials and within the expected stage (stage 3). Table 6 reports the average values and their double standard deviation at the time of the detection. Similarly to the S5 results, a small variability of the angle was found (6.2 %), whereas the distribution of the torque and of clamping force was in a wide range (377 % and 447 %, respectively).

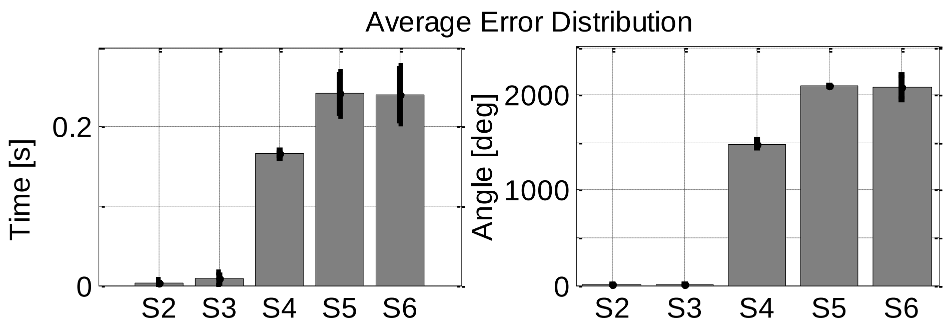


Figure 36: Average distribution (in grey coloured bars) and two times standard deviations (black lines) of the time and angle over all the different error scenarios S2 - S6 (left and right panels, respectively) based on Table 5.

In summary, Figure 36 shows the average times and angles for all error scenarios which have been reported before. The figure shows when and where the errors were detected in terms of time and angular position, respectively.

3.6 Conclusion

This chapter investigates the bolt tightening assembly in the framework of wind turbine hub assembly. The wind turbine hub contains up to 128 bolts which are used to mount the bearing onto the wind turbine hub. The assembly process requires to be completed with high accuracy level.

The process under investigation is non-linear and suffers from uncertainties (such as variations in friction, angle, environment and material). Errors need to be promptly detected at an early stage to avoid any damage and to ensure that the assembly is completed according to the requirements and specifications.

To address this issue, a model-free FLC has been designed and implemented, based on a physical system analysis. According to the analysed uncertainties, such as the variations of frictions and angles, the tightening process has been subdivided into 4 stages to include specific knowledge about the tightening operation for each of the stages and also error recognition so that the FLC can return an error feedback incorporating information about the occurring stage.

The FLC implemented on a real time industrial control system ensures that the spinning speed, during the critical times of the process, is reduced and the tool runs on the required specifications and proves to be able to reach a specific clamping force.

Results have been compared with an industry standard PID controller. It has been shown experimentally that the new 4 stages FLC performs better overall, as its error is lower and it provided higher and constant accuracy as well as it is more robustly..

CHAPTER 4

4. Chapter 3: Model-Based Self-Tuning PI Control of Bolt-Nut Tightening for Wind Turbine Bearing Assembly

Chapter 4 outline

Outline – The four stages fuzzy control strategy introduced in the previous chapter is able to perform the tightening process on a high reliable performance level. The difficulty of the bolt/nut system is the uncertainty and nonlinearity of the bolt system. Models can be derived theoretically or experimentally. Theoretical models may be difficult to form in a way that they reach the required accuracy and match the physical system. An experimental model may be more suitable as it can include a certain level of nonlinearity and uncertainty.

This chapter shows a two stage control strategy, where a numerical model is derived experimentally and is used for PI tuning. At the same time, the derived gains may not be accurate as the model is not fully matching the physical system. The gains are then further fine-tuned by an online genetic algorithm which is directly applied to the physical system and learns its individual attributes. Therefore, the performance increased to a very reliable level. Compared to the previously introduced error detector based on fuzzy, the introduced error detector is based on a logic table.

The two stage controller has been implemented within an industrial PLC (Programmable Logic Controller) based on a Phoenix Contact control system. It has been successfully validated and compared with the performance of other control strategies, such as model free approaches. It turned out that the tightening has been completed with high accuracy and all errors were successfully detected.

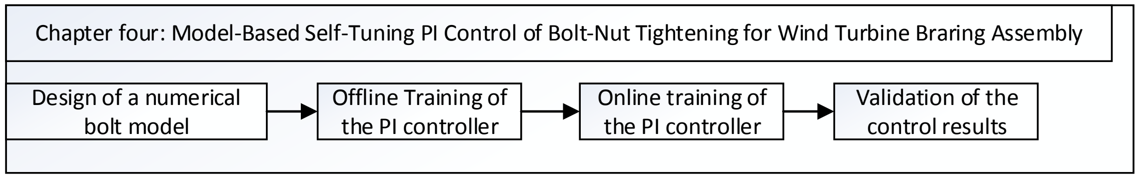


Figure 37: Chapter three structure

4.1 Model-free control strategies

Model-free control strategies can be achieved using fuzzy or neural network controllers, as described in [21] [51] [29] [52] - [62] [63]: with these controllers, an expert knowledge of the process is inherited within the plant to address all non-linear ties and error possibilities; some approaches presented in the literature even have the ability to detect errors occurring during the tightening [21]. The advantage of such an approach is that high accuracy can be achieved without the need for an explicit numerical model of the physical system [21], which models all the non-lineareties and uncertainties.

One ideal control strategy for completing the tightening process is the introduced four stages fuzzy control tightening technique. It is a not based on a numerical model and the control target can be reached using membership functions and linguistic rules which incorporate knowledge to the bolt system and error scenarios [21] [11] [16] [29].

PI control is one of the most popular techniques that can be used for the automation of the tightening process and it usually performs well on linear systems. Nevertheless, a bolt-nut connection system contains a linear component, which can be modelled as a spring system as well as highly non-linear components, which are difficult to include in a numerical model. In addition, the bolt system contains many uncertainties, caused by the presence of static and dynamic friction occurring between the bolt and nut threads during the tightening. The non-linearity and uncertainty are normally similar for a set of bolts coming from the same package or manufacturer, as they may have been manufactured on a comparable level (leading to similar friction) and stored together (same temperature and environmental conditions). A different set of bolts may have different attributes and may require an adjusted tightening strategy.

The target for the tightening process is to reach a specific clamping force level which is directly depending on the angular tightening level. To perform this strategy, angle-based tightening methods have been applied [16], where the nut is run down the bolt until the former one touches the flange and is tightened to a desired angle level. The angle depends on the assembly specifications and on the involved coupling materials. According to this approach, many control strategies have been developed, which are based on state space control [11] or PI controllers. However, because of uncertainties including the environmental temperature effect, and of the presence of potential mechanical defects of the bolt [23] [24] [25] - [27], a typical PI controller, whose P and I gains are set to constant values, will have difficulties to perform with high accuracy in this case [28]: the control target might be reached [64], but the non-linearity of the bolt/nut connection may prevent the controller from operating optimally [21], as the control target may not be reached with the required accuracy.

In order to take advantage of these different approaches, this chapter presents a model-based PI control strategy, where the P and I gains are estimated using a numerical model describing the bolt nut system. Then the gains are combined with an online learning Genetic Algorithm (GA), where the values are fine-tuned to handle the impact on the controller by the non-linearity and uncertainties of the physical bolt/nut tightening system. Because an essential requirement of the wind turbine hub assembly is to recognize tightening errors before they can cause any damage [21], a logic based algorithm which runs in addition to the controller is implemented: this logic-based algorithm is based on the real-time monitoring of the torque and angular values which allow detecting several error scenarios such as 'missing nuts', 'missing washers', 'wrong bolt sizes', and 'different thread types'. Whenever an error occurs, the system automatically switches off and reports the error typology to the operator, to allow rapid correction of the error and ensuring the continued execution of the assembly process.

It is noted that the fuzzy based error detector, which has been introduced in the previous chapter, implements the functionality using membership functions and linguistic rules. In contrast, the error detector implemented for the GA PI control strategy is based on a logic based algorithm, which is using comparators with different experimentally derived values describing an error. This error detector runs in parallel in real time to the PI controller and switches off the output of the PI controller in an error event. It should be pointed out that the error detector of the four stages controller is gently decreasing the output of the controller to slow down the tool before the error event occurs. This minimizes the remaining kinetic energy in the tool and therefore the risk of physical damages.

4.2 Control Architecture, System Identification and PI Self Tuning

To provide an accurate control architecture, the bolt-nut connection needs to be modelled in such a way that that the controller can cope with several uncertainties. To achieve that a model of the system should be designed, taking into account the behaviour of the nut on the bolt. At first, a preliminary model is estimated from experimental data of bolt tightening, and then a set of PI gains is estimated based on this model. Since the model does not properly incorporate all uncertainties, a GA is then introduced to fine tune the PI gains: this second phase is performed online, where the initial model-based PI gains are fine-tuned by setting a boundary limit around them in which the GA estimates the best gains (see Figure 38). The control target is to run the nut to an accurate angular level in order to achieve a specified clamping force level. This double stage set-up, where in the first stage the PI parameters are identified on a model and in the second stage the PI parameters are further fine-tuned using an online GA, has the clear advantage that precise control values can be estimated based on the combination of a general PI values estimation added to an online GA estimation process, which adapts and refines the optimal parameters for each set of bolts. It is noted that the each nut/bolt interface introduces different uncertainties and non-linearity.

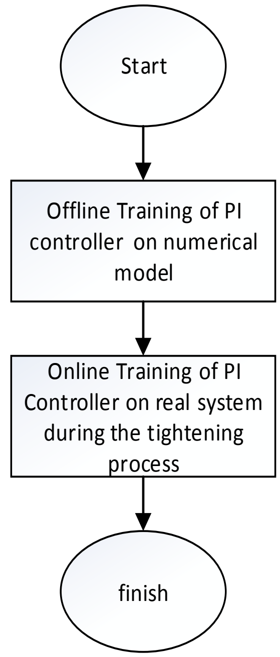


Figure 38: Flowchart of the control strategy and training process

4.3 System analysis and identification of the M24 bolt system

An experimental scenario has been prepared in order to emulate a semi-automatic wind turbine assembly procedure: a tightening tool - model DSM MDW140, DSM Messtechnik Germany - is mounted on an industrial robot arm (Fanuc M6-iB) which moves the tightening tool from one bolt to the next on the wind turbine hub (Figure 39). A bolt bench or customized clamp is employed to prevent backwards torque being transmitted into the robot arm: the clamp is attached to the robot and grabs onto the flange. The tightening tool integrates a torque sensor and an encoder, allowing to monitor the torque applied to the nut and its angular displacement as a function of time, respectively. A magnetic socket at the end of the tool allows picking M24 nuts and tightening nuts on the respective bolts (M24 bolts and nuts are commonly used in this type of application). To measure the effective and final clamping force, the system is integrated with a clamping force sensor, model MecSense AST 50KN (Figure 40). The effective clamping force settles after completing the tightening process to the final clamping force.

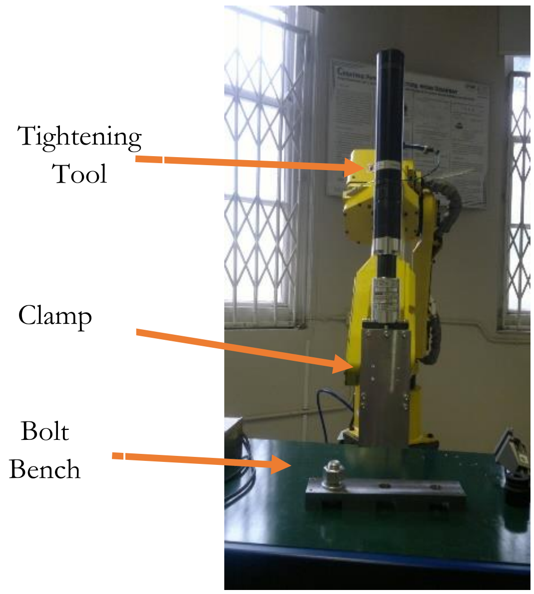


Figure 39: Fanuc Robot with the bolt tightening tool

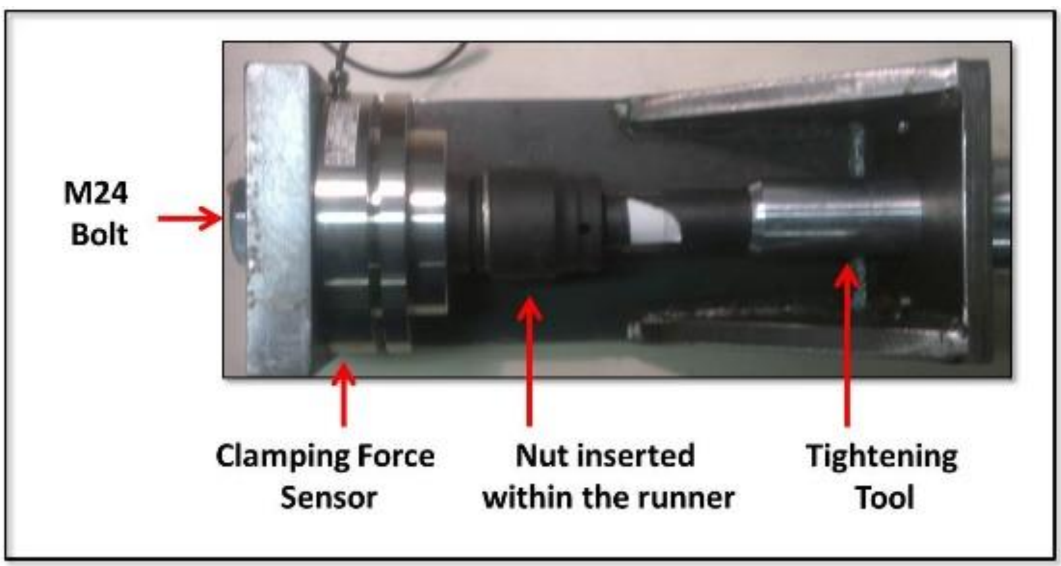


Figure 40: Experimental bench with the clamping force sensor, the M24 bolt and the tightening tool

To estimate the bolt system model, the servo drives of the tool and gearbox and the bolt as well as the nut need to be included in the model design. To characterize a general state-space model, the MATLAB System Identification Toolbox (the Mathworks Inc.) has been used. The final target of the controller is to drive the tool to a desired angular nut position with a desired clamping force on the flange, where the bolt-nut tightening occurs, as it is typically specified within the assembly requirements for each individual assembly item.

In order to identify the numerical model describing the input (Voltage as speed signal) and output (tightening angle), a sine wave voltage signal is applied to the tool as soon as the nut is engaged with the bolt. The same sine wave in the form of

$$\mathrm{v} = \mathrm{A} * \sin \mathrm{b}(\mathrm{t}) \qquad (3)$$

is applied 100 times (1 Hz, 4V) to drive the nut up and down along the bolts thread while recording the system response in terms of angular displacement. The operating range of the angular displacement has been set between 0° to 70°, where the zero position is at the point where the nut touches the flange. At this stage, it is important to observe that the clamping force is not represented within the model, as the desired clamping force is directly related to the angle. The clamping force cannot be permanently measured in the assembly line (the Sensor in Figure 40 is only used in test environments)

The experimental data are then used and the numerical model is identified by assuming a linear system model:

$$\frac{\mathrm{dx}(\mathrm{t})}{\mathrm{d}(\mathrm{t})} = \mathbf{A}\mathrm{x}(\mathrm{t}) + \mathbf{B}\mathrm{u}(\mathrm{t}) \qquad (4)$$

$$\mathrm{y}(\mathrm{t}) = \mathbf{C}\mathrm{x}(\mathrm{t}) \qquad (5)$$

where $\mathbf{A} \in \mathbb{R}^{8\mathrm{x}8}$ is an 8x8 matrix equal to:

$$\mathbf{A} = \begin{bmatrix} 0{,}001006 & -0{,}0070 & -0{,}0325 & 0{,}01128 & -0{,}0112 & -0{,}011 & 0{,}01547 & 0{,}034 \\ -0{,}0294 & 0{,}2774 & -35.415 & 0{,}1921 & -3{,}8959 & -2{,}8667 & 1{,}4899 & 6{,}7001 \\ -0{,}0093 & 32{,}3713 & -2{,}579 & 4{,}7173 & -5{,}991 & -2{,}3723 & 5{,}9476 & 5{,}605 \\ -0{,}0156 & 0{,}1534 & -1{,}9704 & -0{,}2615 & 16{,}0768 & 4{,}8054 & -2{,}5879 & -2{,}8243 \\ -0{,}0056 & 0{,}6292 & 1{,}1228 & -11{,}363 & 0{,}235 & -10{,}031 & 3{,}9161 & 12{,}914 \\ 0{,}0503 & 0{,}7567 & -0{,}6807 & -2{,}6266 & 8{,}6707 & -0{,}7461 & 3{,}5759 & 7{,}603 \\ 0{,}0385 & -0{,}7156 & 1{,}332 & 1{,}3684 & -0{,}351 & 0{,}7911 & -6{,}8495 & -70{,}43 \\ 0{,}0569 & 0{,}5129 & -0{,}2292 & 0{,}2269 & -1{,}5451 & -2{,}2602 & 43{,}5787 & 4{,}9514 \end{bmatrix}$$

$\mathbf{B} \in \mathbb{R}^{8x1}$matrix:

$$\mathbf{B} = \begin{bmatrix} 1{,}35069 * 10^{-7} \\ 1{,}16539 * 10^{-4} \\ -2{,}07081 * 10^{-5} \\ -7{,}18585 * 10^{-5} \\ 1{,}98291 * 10^{-5} \\ -4{,}307479 * 10^{-5} \\ -0{,}0002156 \\ -0{,}000173 \end{bmatrix}$$

$\mathbf{C} \in \mathbb{R}^{1x8}$matrix:

$$\mathbf{C} = [-0.0001 \quad 5.2131 \quad -5.8923 \quad -7.2507 \quad 1.5741 \quad 0.6202 \quad -5.5276 \quad 1.2935]$$

The B Matrix shows rather low values, which, as a consequence, cause high P values for the control strategy, however, the real system is physically limited with regards to the range of its input values, and hence the analog input voltage is limited. Therefore, for any PI estimation method, this needs to be considered before the control strategy is finalised.

The result of the simulated model compared with real experimental data shows a fitting accuracy of 95.04 %. This matches the system quite well, however, the model does not cover the uncertainties and non-linearity.

Although the size of the model increases with its order, it is found that the experimental data is matched best with an 8th order state space model. It is noted that, since the real assembly system has variations in friction between the nut and the flange as well as between the threads, the model needs to be identified for each assembly system (i.e. the tightening tool) and for any bolt size.

It needs to be considered that the model may change if a different type of bolt has to be used; hence attributes such as the friction, the tension limit and the desired angle range may change. This constraint clearly implies that the model has to be re-identified and re-designed for each size of the applied bolts. The tension limit is the maximum tension force, which can be applied to the bolt before it reaches its plastic region [16]. The approach also considers error recognition. Bolts can be damaged, have the wrong stiffness, wrong threads or sizes with regards to the used nuts. It is essential that such issues are detected before the assembly is complete. Ideally, the tool should stop immediately when an error occurs to prevent any physical damage. Further error scenarios have been defined in the validation section.

4.4 Applying the genetic algorithm

The identification of the general bolt model leads to the second stage process based on a genetic algorithm (GA), employed to adapt and optimize our PI control loop strategy (see Figure 41). The PI approach has been chosen for its simplicity, as only two gain parameters need to be tuned. However, any other control strategy (e.g. state space control) can be used and, as demonstrated here, the parameters can be fine-tuned using a GA.

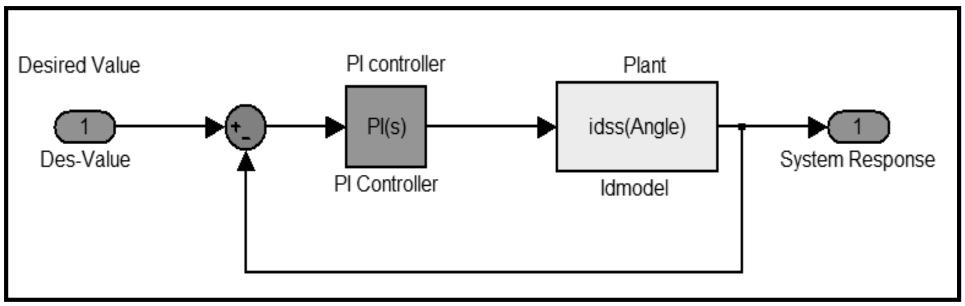


Figure 41: PI control loop

Within the control loop the 'stage 1' angle-model (i.e. the 'Plant' block) has been integrated in order to simulate the system response of the tightening tool. Then, a standard GA [16, 32] is implemented in real time which optimizes the step response of the PI controller subject to a performance index, which is defined as follows:

$$J = \int_0^{t_f} \mathbf{x}(t)^T \mathbf{W}\mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t) dt \tag{6}$$

where is the system state, $\mathbf{u}(t) \in \Re^m$ is the control signal, $0 \leq \mathbf{W} \in \Re^{n \times n}$ and $0 \leq \mathbf{R} \in \Re^{m \times m}$ are user-defined weighting matrices, $t_t > 0$ is the time period of the optimization taking place. By minimizing the performance index J (or the fitness function using GA terminology), the energy consumed by the system states and control signal is minimized according to the weights specified in **W** and **R**.

The GA-based optimization process is shown in Figure 38. In the first step, the numerical bolt model is used for training, where the PI gains are at first generated randomly. Once this is completed, the PI gains are further fine-tuned on the physical tightening system. Then, the

PI-controlled system is evaluated by using the performance index J, which has been reported in Eq. (6). The GA evolves the PI gains in the next iteration aiming to improve the control performance measured by the performance index J. This process is repeated until a stopping criterion is met. The criterion is defined by a further improvement of the step response and is fulfilled when it is dropping below a performance level, where a further improvement of the control strategy is not possible.

The used real-coded GA is standard and uses roulette wheel selection (cost weighting selection), arithmetic crossover and non-uniform mutation. The chromosome is chosen to be [P, I], the lower and upper bounds of P are -100 and 100, respectively; the lower and upper bounds of I are -10 and 10, respectively; the population size is 20; the probabilities of crossover and mutation are 0.8 and 0.05, respectively. The number of iterations is 500. The genetic algorithm stops when the maximum number of iterations has been reached. The optimal values of P and I are found to be P = 99.999 and I = 4.7244, respectively, with the following weights: $\mathbf{W} = \mathrm{diag}\{1, 0, 0, 0\}$ and $\mathbf{R} = 30$ as shown in Formula 6.

4.5 Error detector

The error detection function of the controller is implemented as follows: in addition to the PI controller a logic-based rule table is implemented performing two main functions, (i) run down the nut to the flange and (ii) initiate the tightening PI algorithm – with the main aim being to minimize the risk of mechanical damages. Different types of errors are defined, according to real bolt tightening scenarios, including misalignment of the nut, missing nut, damaged thread of the bolt or nut, and wrong coupling between the nuts and the bolt (or different thread type). Table 7 lists the defined error types and the typical behaviour of the torque and angular patterns for each case. It is noted that the torque and angular conditions are related via an AND logical operator, meaning that, for example, a misalignment error scenario is detectable when the torque value is more than 10 Nm AND the angular value is less than 15°. More error types can also be integrated by extending the logic of the table, as reported in [49].

Error Type	Torque level	Angle level
Nut misalignment (jamming)	high (above 10 Nm)	low (below 15°)
Missing nut	low (below 5Nm)	high (above 50°)
Damaged bolt (cross thread)	high (above 10Nm)	medium (15° - 47°)
Different thread types	high (above 10Nm)	low-med (0°- 47°)
Damaged nut thread	high (above 10Nm)	low-med (0°- 47°)

Table 7: Error detection values; an error flag is set based on comparators for the set values

The values for error detection (both for torque and angular level) depend on the assembly specifications and are based on the control target and the bolt size.

4.6 Validation of the control strategy

As described in 3.3, the system set-up of Figure 40 has been integrated with a clamping force sensor, to validate the model. The set-up contains an M24 bolt and the tightening tool; the test rig is attached to a bench and accommodates the recording of the clamping force (see the Figs. 3 and 4).

To validate the controller in an industrial environment, the two-stage control architecture and error detector have been implemented using the MATLAB Programming Language (the Mathworks Inc.): a Simulink model has been prepared and implemented with the MATLAB Coder, to generate a real time C++ code. The code has been imported into a Phoenix Contact Industrial Control system [59] and executed in real time with a cycle time of 1ms: relating to the set-up shown in Figure 39 and Figure 40, the nut has been positioned at the end of the bolt and is then run down to the flange with the tightening tool, to bring it to a specified final position where it touches the flange. The PI controller is switched on at the time when the nut touches the flange, which is defined as 0°.

A first set of tests has been performed at three operating points, namely at 28°, 40° and 45°. Ten trials have been conducted for each one of the operating points and a statistical analysis has been carried out. These three operating points have been also simulated and simulation results have been compared with the experimental ones.

A second set of tests has been performed to measure the error detection capability; several error scenarios were set up and successfully detected by the system.

At first, the GA runs on the model in a simulation environment to pre-estimate useable PI Gains which will be further fine-tuned in the 2nd step.

4.6.1 Response of the system with deactivated GA stage

4.6.1.1 Tightening angle set to 28°

As a first operating point, the angle has been set to 28°. The control loop has been implemented and simulated within the Matlab Simulink environment and it has led to the following system response as it is shown in Figure 42.

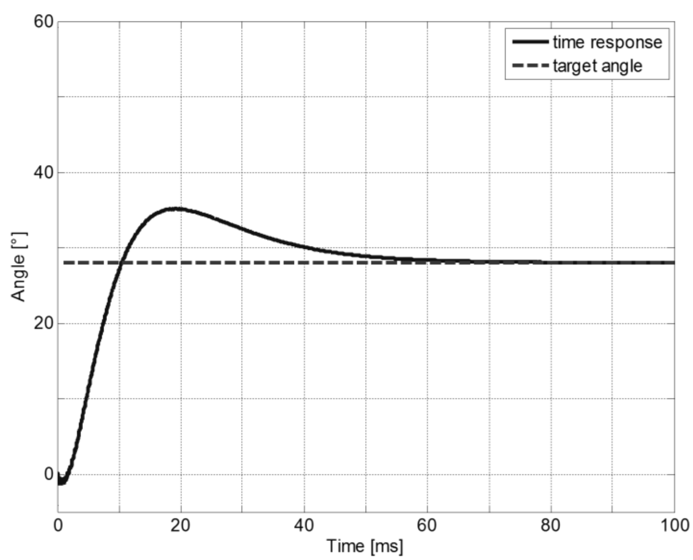


Figure 42: Simulation result for a tightening desired angle of 28°

At this point, the same target angle of 28° has been set-up on the real tightening system and the system response has been recorded. This leads to the behaviour reported in Figure 43.

A time drift can be noticed between the two plots due to the fact that, while the simulation plot begins at t = 0 s, the experimental one starts after the acquisition of the data was switched initialized (i.e. t = 0.49 s). Therefore, the initial condition is the same in the simulation.

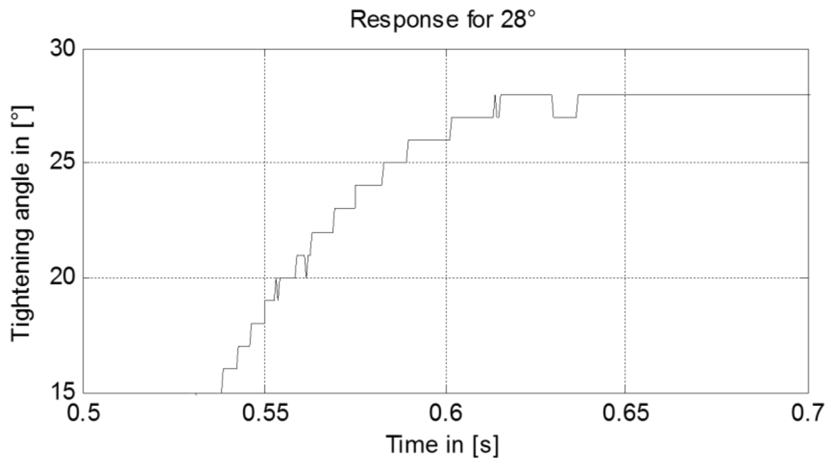


Figure 43: System response for a tightening desired angle of 28°

It can be noticed that no overshoot is observable in the experimental results. This is due to the mechanical limitations of the tightening tool: in fact, the control signal became too small to be able to drive the tool against the internal friction of the gearbox, which is the friction between the nut and the flange, of the servo drive and of the tightened bolt. Therefore, it stops at this point and the angle settles.

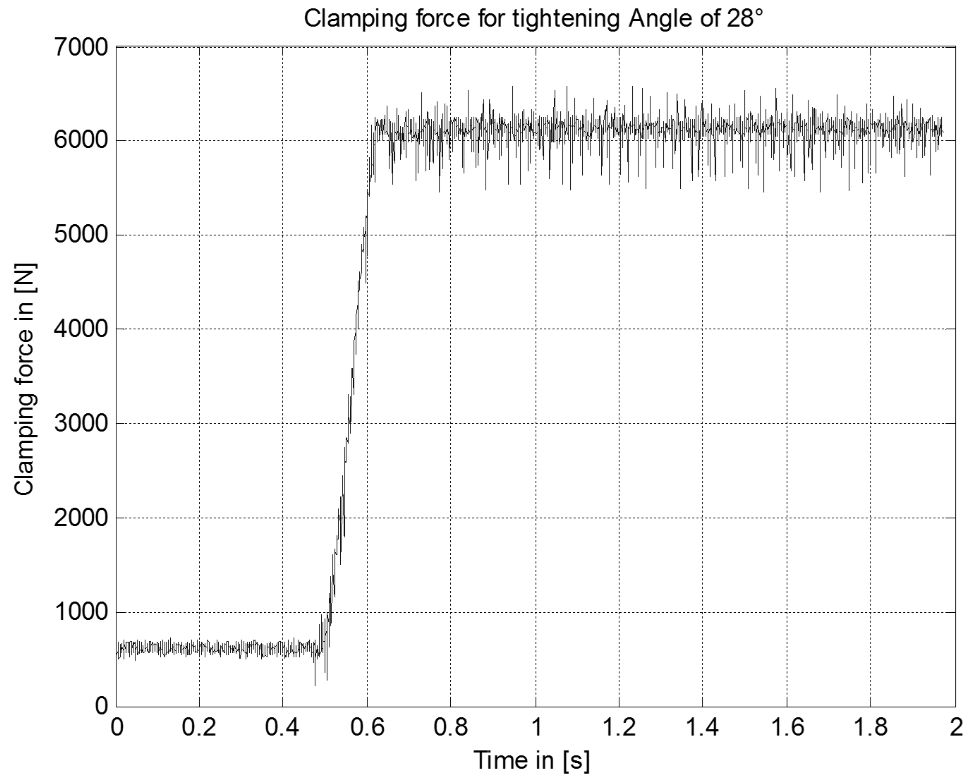


Figure 44: Clamping force for a desired tightening angle of 28°

The clamping force has also been recorded during the experiment and it is shown in Figure 44. It is starting from a low level, which is caused by the pre-tightening (where the nut touches the flange) - when the nut has been placed and run down to the flange with no specific angle or torque level - and then it is rising in a similarly way of the angle.

Target Angle	Mean	Standard Deviation
28°	27.8°	0.7012
40°	39.75°	0.9
45°	44.2°	0.8367

Table 8: Mean and standard deviations of the PI controller (not GA optimized)

The trials have been repeated five times and the average and standard deviation of the angle and force have been reported as a statistical Gaussian distribution in Figure 45. Noticeably, the graph of the clamping force contains a certain level of noise which is caused by the measurement. The Gaussian distribution shows that the clamping force confidence level is slightly lower than the angle confidence level. This is due to friction uncertainties of the bolt and it also depends on the twist level of the bolt. When a bolt is tightened it gets slightly twisted which has an effect on the clamping force.

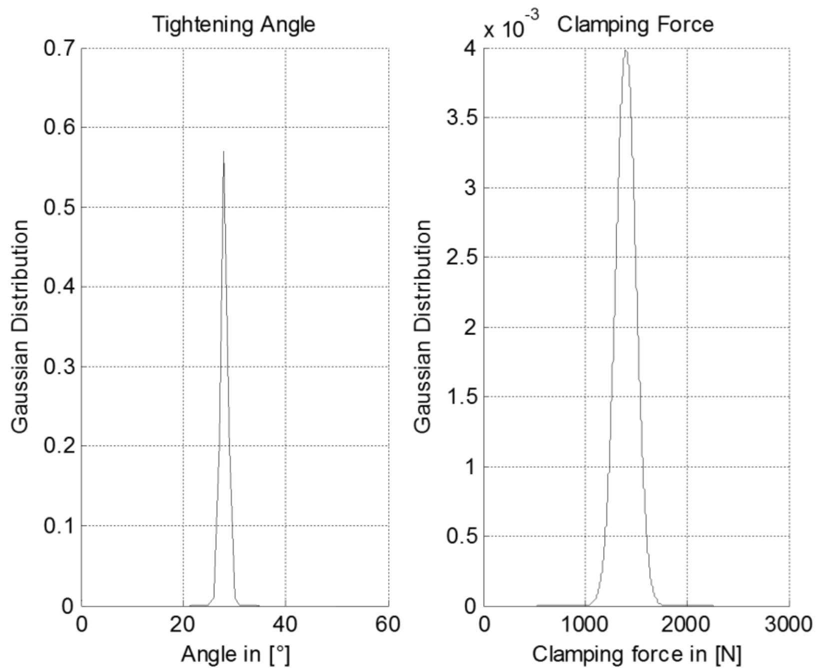


Figure 45: Gaussian distribution for all experiments for a desired tightening angle of 28°

The same experiment has been carried out at the operating points of 40° and 45° with very similar and consistent results.

4.6.1.2 Tightening angle set to 40°

The initial simulation result for 40° is shown in Figure 46 and it displays an overshoot of about 10°. The physical controller's desired value has been set to 40° and the experiments have been re-executed. The angular and clamping force time patterns are reported in Figure 47 and Figure 48, respectively. Similarly, five trials have been performed and their statistical analysis is graphically shown in Figure 49.

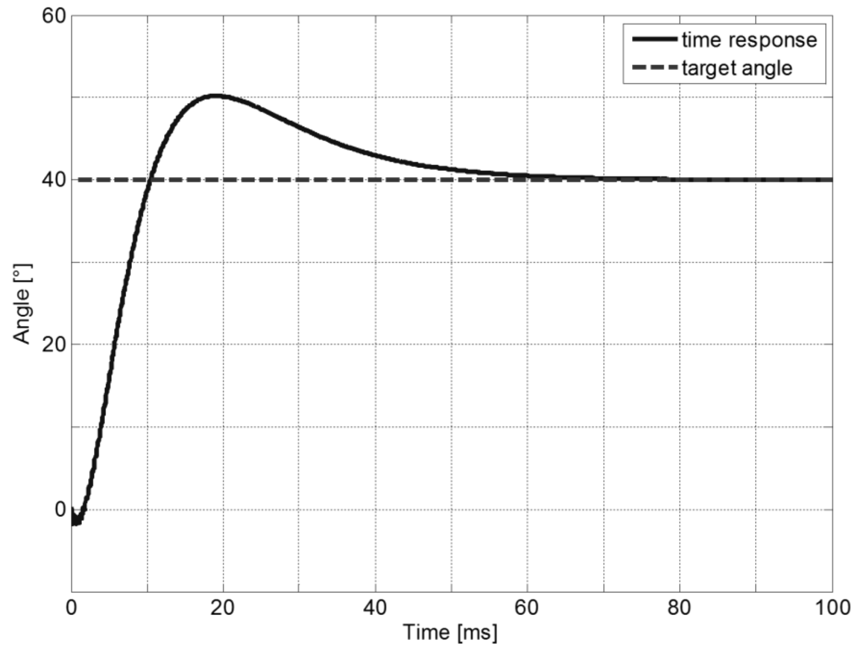


Figure 46: Simulation result for a desired tightening angle of 40°

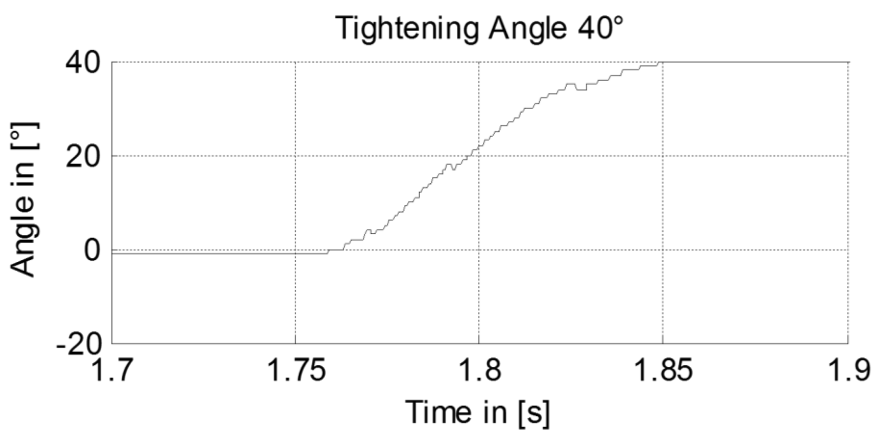


Figure 47: System response for a desired tightening angle of 40°

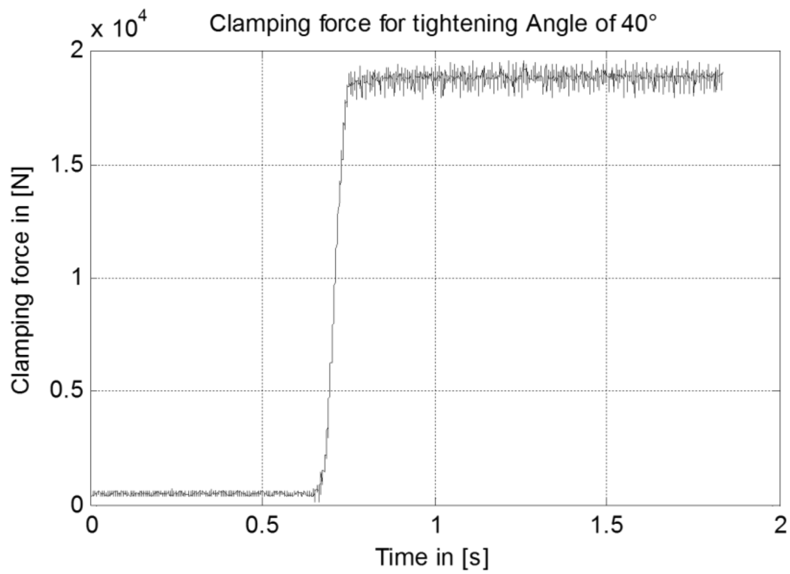


Figure 48: Clamping force for a desired tightening angle of 40°

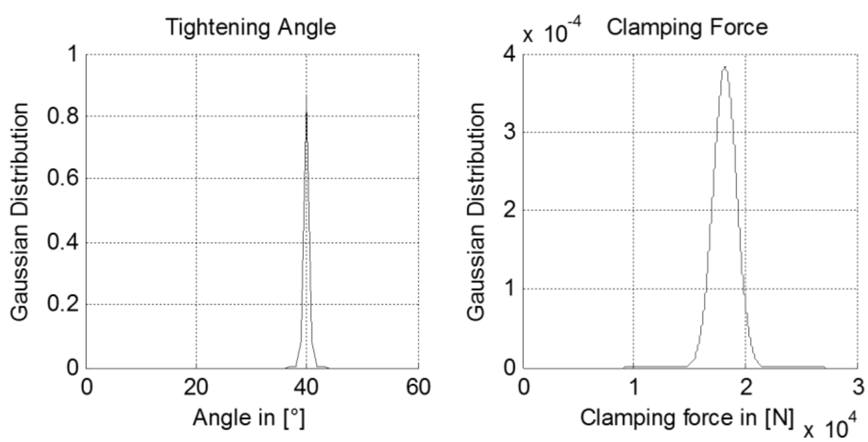


Figure 49: Gaussian distribution for a desired tightening angle of 40°

4.6.1.3 Tightening angle set to 45°

Finally the same approach has been followed on the third operating point, namely at 45°. Five experiments have been carried out with comparable outcomes, as it is shown in Figure 50, Figure 51, Figure 52 and Figure 53.

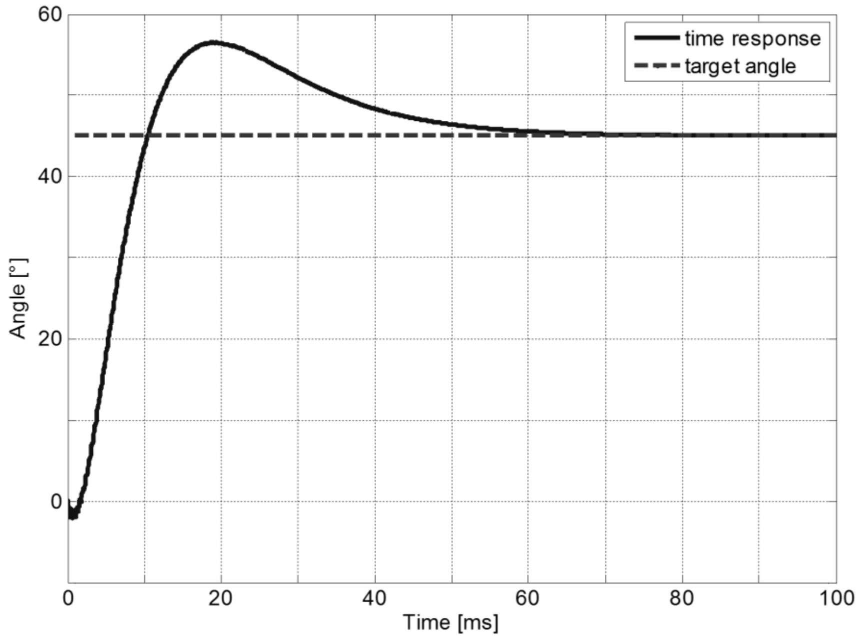


Figure 50: Simulation result for a desired tightening angle of 45°

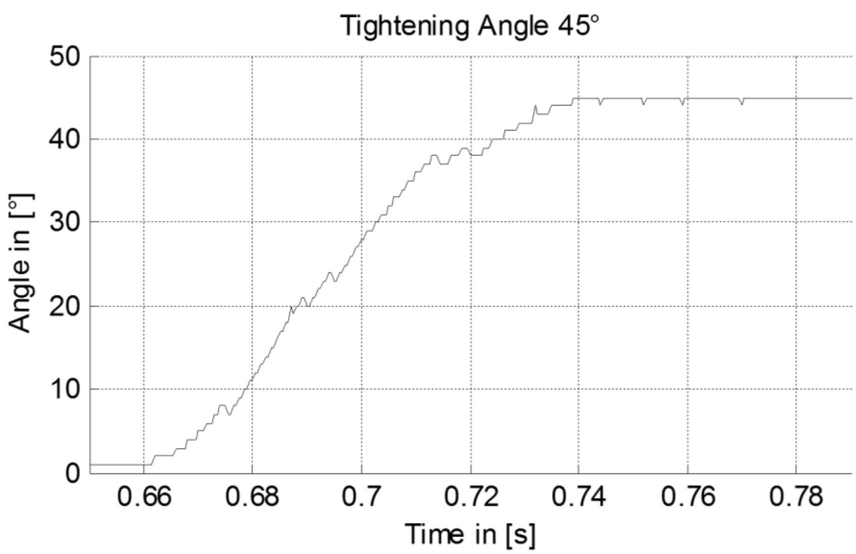


Figure 51: Step response for a desired tightening angle of 45°

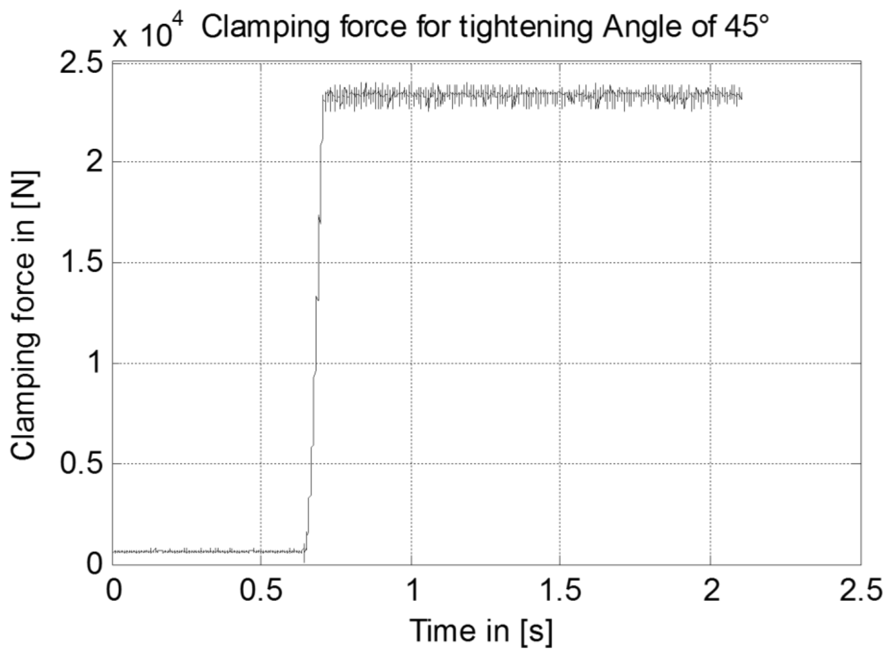


Figure 52: Clamping force for a desired tightening angle of 45°

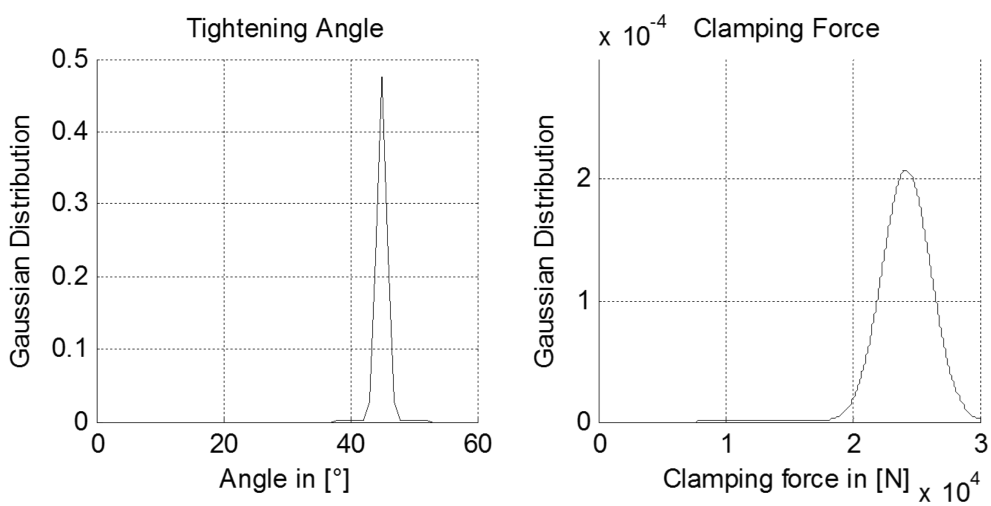


Figure 53: Gaussian distribution summarizing all experiments for 45°

4.6.2 Combining the genetic algorithm with the PI controller

The model is not accurate since it does not include all non-linear components and uncertainties of the system. This can be overcome by using a GA online algorithm, which further analyses the physical system. The previously estimated PI values have been included in the PI algorithm and the boundaries have been set to ± 30 at the P value as well as ±15, which have been identified experimentally.

For the real system the GA has been modified in a way that the PI values match the physical limit of the tightening tool. The tool is voltage driven which sets a speed signal to make the tool spinning. Therefore, the maximum P value has been estimated at 130 and I value boundary has been set between 0 and 20 (a negative I value turned out to stop the tool from working accurately). Furthermore, the GA has been applied and it returned a result of P= 113.82 and I= 10.11. This is within the boundary set around the gains which have been estimated in the model.

The results have been tested on the physical system and with different values of the desired tightening angle. A new experiment has been carried out to outline the performance on a different operation point which is still in the operation range of the defined model. One sample result is shown in the figure below, targeting an angle of 31°:

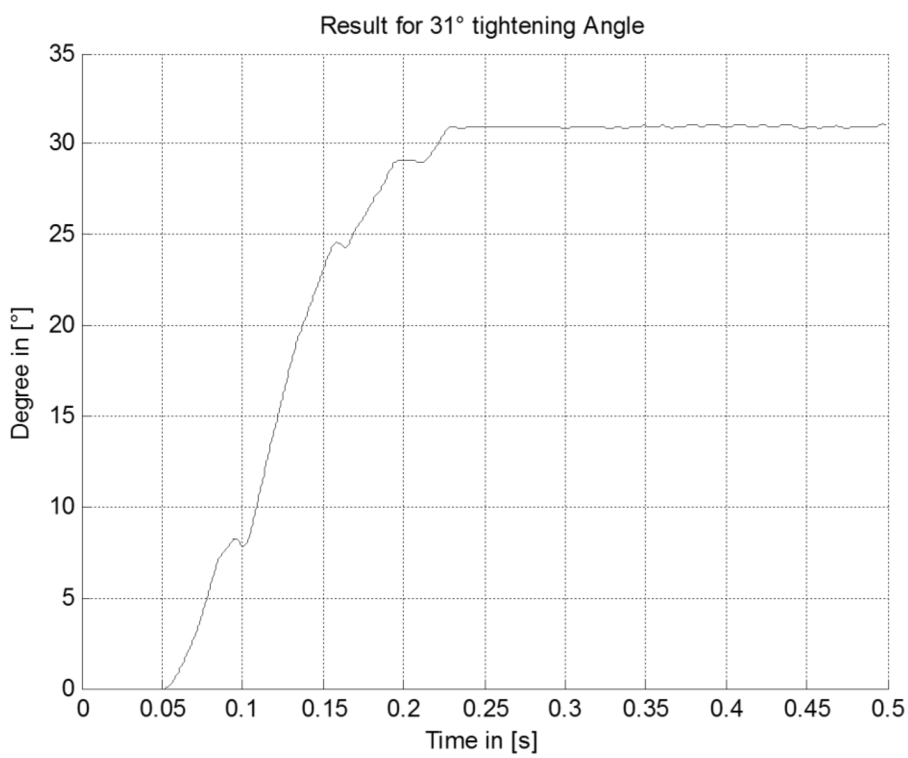


Figure 54: Tightening angle for a desired value of 31°

The oscillation at the end of the step response is caused by noise on the system. It shows that the control strategy can also be used on various angle levels which are within the operating range of the model.

Summarizing, the Gaussian distribution for this angle has also been estimated based on 100 Experiments, as shown in Figure 55.

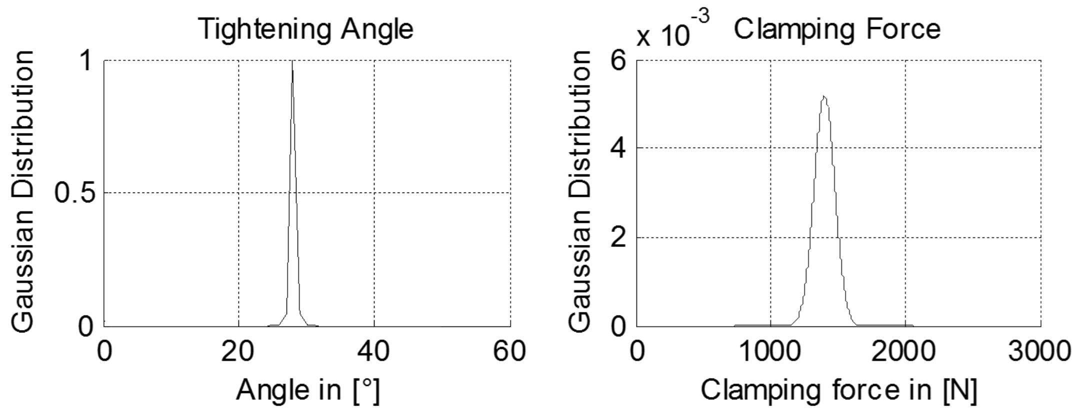


Figure 55: Gaussian Distributions for a desired tightening angle of 31°

As it can be seen in the Gaussian distribution, accuracy has been significantly increased compared to the previous experiments where the results are entirely based on the estimated gains based on the numerical model.

Concerning the error detection capability of the algorithm, the logic based algorithm which is on top of the PI Controller, constantly compares the torque and angular values with defined thresholds describing error scenarios, as shown in Table 7. Five different error scenarios can be automatically detected: nut misalignment, missing nut or jamming, damaged bolt or cross thread, different thread types and damaged thread of the nut. A 'misalignment' condition is always detected when the measured angle is lower than 15° and the torque amount is more than 10 Nm. Similarly the 'missing nut ' condition is promptly detected as soon as the torque value is lower than 5 Nm and at the same time the angular position is greater than 50° and so on. All the threshold values of the torque and angles have been predetermined by means of preliminary trials in which the diverse error conditions have been tested.

Another error type is a wrong stiffness of the bolt or a damaged bolt which may not be able to provide the desired clamping force. This can be detected by estimating the deviation of the torque during the run down – if it stays constant the tension limit of the bolt has been reached. If the desired angle has not been reached at that time, the bolt is not able to provide the clamping force, which means that it is either structurally damaged or has the wrong stiffness. For testing, the error scenarios have been set up and 20 experiments have been carried out.

Error Type	Torque level	Angle level
Nut misalignment (jamming)	All trials above 10Nm	All trials below 15°
Nut missing	All trials below 5Nm	All trials above 50°
Bolt damaged (cross thread)	All trials above 10Nm	All trials between 15°-47°
Different thread types	All trials above 10Nm	All trials between 0° and 47°
Nut thread damaged	All trials above 10Nm	All trials between 0° and 47°

Table 9: Error detection results

The error detector has been tested and it turned out that (as expected) it works with one cycle delay. The Phoenix Contact system executes the program cyclically with 1ms, meaning that all errors have been detected within 1ms and the system switched off to prevent any physical damage. When an error is detected, a flag is set which stops the PI Controller. Once the error situation is resolved the flag can be reset and the tightening process can continue.

4.7 Discussion

The presented GA can be used for the design of a well performing PI controller. The bolt-nut physical system has been modelled as a state space model (step 1) and then used to estimate an accurate set of PI values of the controller based on the GA (step 2).

The operating range of the model has been defined from 0° to 70°: the tightening tool has been placed on an M24 bolt and a sine wave applied to estimate the system response for as many input values as possible. During this estimation, the nut has not touched the flange and ran on the bolt shaft only (since otherwise, due to the tightening tool performance limitations, the angle range could not have been covered). The actual performance of the model based controller has been validated in three operating points, namely at a desired tightening angle of 28°, 40° and 45° The online fine-tuned GA based controller has been tested on a new target angle of 31° where it showed a significant improvement.

The GA has been prepared for this application and the results have shown that 100 iterations estimate a more accurate set of PI value than the original model based estimation. It turned out that the introduced number of iterations is sufficient to correctly estimate the set of the PI values. Further iterations may improve the PI values further, but the mechanical constrains of the tool will not take advantage of that. The values have then been used in a discrete PI controller within a Simulink model and converted into real time C++ Code to be imported into a Phoenix Contact industrial PC for the real time execution.

The controller has been simulated using Simulink and the results have shown an overshoot of the desired value. When the same angle has been set to the physical tightening tool it turned out, that the overshoot was rather low compared to the simulation. This is due to the mechanical limitations of the tool, since its servo motor is driving both an integrated gear-box and the actual nut, which is running on the flange (introducing additional friction). This requires a higher input signal but, as soon as the controller is getting close to the desired value, the input signal becomes too low to cause the overshoot and the system settles at the desired value.

Figure 51 shows an oscillation which is also due to the fact that the input voltage of the tool is low as the control target has been reached but still high enough to move the tool where it causes this oscillation. Furthermore it also needs to be considered that the nut and the nut-runner have a little bit of mechanical play. The nut has been tightened to the desired 45° and the tool oscillates by 1° but, as soon as the nut has reached the desired angle, it will stay there since there is friction between the flange and the nut. In fact, the 1° oscillation is within the play range of the nut-runner.

The obtained clamping force can be reached within specific thresholds and has a lower confidence level than the actual tightening angle as it can be seen in the Gaussian distributions (Figure 45, Figure 49 and Figure 53 show the fine-tuned result which was not achieved online, while the final online result is shown in Figure 55). This is due to the uncertainties of the friction between the bolt's and nut's threads as well as to the uncertainties of the friction between the nut and the flange. The Gaussian distributions also show that the range is within an acceptable threshold.

In [21] a similar problem is analysed by using a model-free fuzzy control application approach. By comparing the results of this paper it can be seen that the PI controller provides a higher performance after GA tuning. The PI controller now includes most non-linearity and uncertainty. However, if a new set of bolts is introduced the gains may need to be fine-tuned again.

4.7 Summary

This chapter introduces an innovative way to address the highly important assembly problem of the wind turbine bearing.

The proposed approach uses a GA in order to estimate the best PI control parameters. In order to do this, a detailed state-space model has been derived using MATLAB System Identification Toolbox and the algorithm has been applied to the state-space model. Once satisfactory values have been found the controller has been simulated for three operating points and also tested on the physical bolt system. It turned out that the derived PI values provide an acceptable performance. This has been verified by doing several experiments for each operation point and summarizing them in a statistical analysis.

The described approach is part of a novel learning strategy which can be applied in the wind turbine hub assembly process.

5. Chapter 5: Comparison of the control concepts

Chapter 5 outline

This chapter compares the model free fuzzy control strategy with the model based controller. It shows the advantages of the model-free fuzzy controller which can be used on various other assembly processes which includes bolt tightening processes, such as the aerospace manufacturing industry.

Different error detectors based on fuzzy control and a logic based algorithm are also compared to its performance.

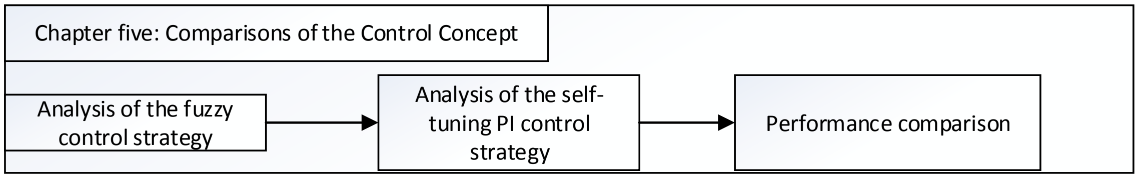


Figure 56: Chapter 4 structure

5.1 Performance comparison

Comparing the performance of the four stages fuzzy controller with that of the model based self-tuning PI controller, it should be noted that the basis of these control strategies is completely different. The PI controller will start when the nut touches the washer at the flange, whereas the four stages controller will start when it is placed on top of the bolt. The Gaussian distributions are compared based on overall performance level.

A model-free control strategy can more easily integrate the non-linearity as well as the uncertainty of the system and provide a constant performance level. The performance of the fuzzy controller is always constant as its parameters are not based on any numerical model.

A model-based strategy based on a PI controller can perform well on a numerical modelled system, and on linear systems in particular. For the bolt tightening application, the control system is non-linear and the performance may not be initially as good as when using the fuzzy controller. By including a genetic algorithm, the non-linearity and uncertainties of a particular set of bolts can be learnt.

Figure 57 shows a comparison between a classical PID control design using the rule of Chien, Hrones and Reswick [65] and a fuzzy control strategy.

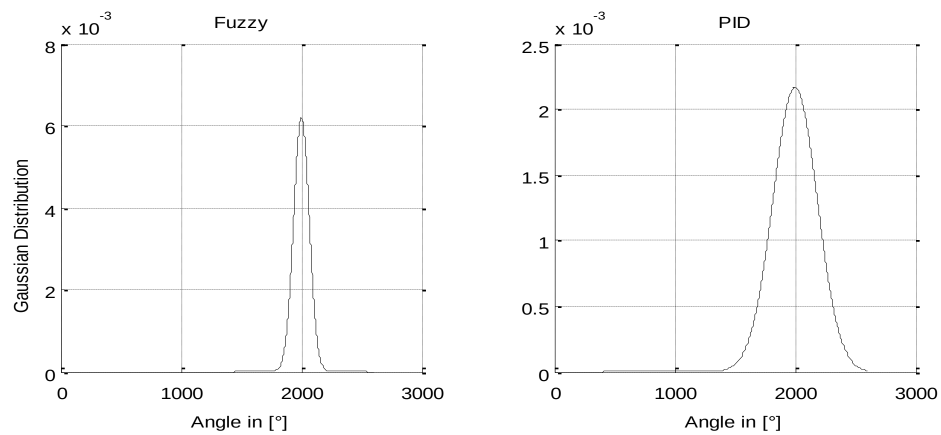


Figure 57: PID vs. Fuzzy

The model-based approach estimation of PI values may provide better gains. However, due to the non-linearity's, the overall performance will initially vary, as the PI gains need to be fine-tuned.

The other tightening control result (Figure 58) for the final tightening angle shows the final angle after training of the Genetic Algorithm. The gains are now fine-tuned and the average distribution of the final tightening angle is now lower than before. This shows that the genetic algorithm adapted to the non-linearity and uncertainty of the bolt system.

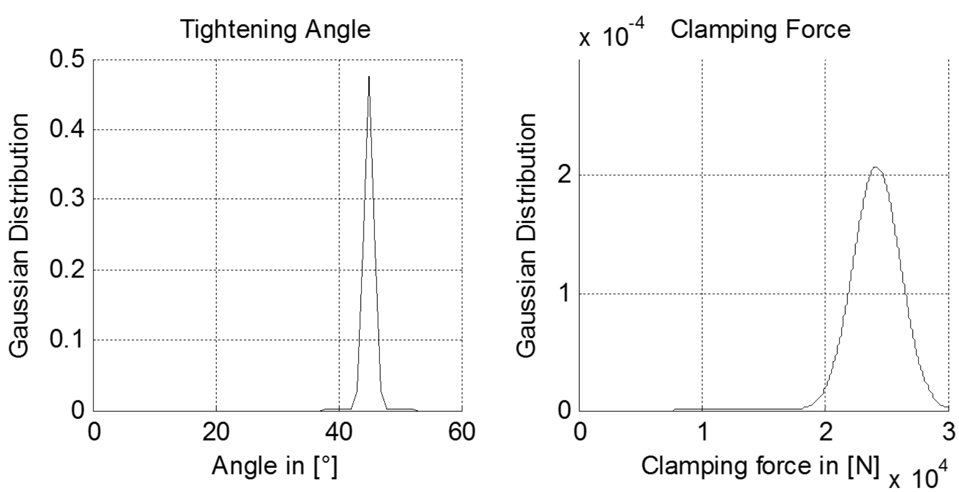


Figure 58: PI fine-tuned Controller using a Genetic Algorithm after training

Comparing the distributions of Figure 57 and Figure 58, it can be seen that the final distribution for the tightening angle is at best for the fine-tuned controller after adapting to the non-linearity of the system. However, there are fundamental differences in each control strategy which need to be considered.

The PID controller shown in Figure 57 is based on a practical approach which is providing approximate PID values (rule of Chien, Hrones and Reswick), the result is not as good as the GA self-tuning algorithm, the performance before the offline training process may be similar to the Chien, Hrones and Reswick approach, since the numerical model for the initial estimation of the PID parameters is not accurate enough.

The high performance shown in Figure 58 will only be reached using the online training which further improves the tightening angle result to the shown level. It needs also taken into account that the bolt system changes if a new set of bolt is introduced from a new package of bolts with different levels uncertainties and non-linearity's. The performance will then drop again to the level before training. The genetic algorithm needs to retune the gains of the PI controller again to reach the optimal performance level.

Lastly, the four stages fuzzy controller is compared with the Chien, Hrones and Reswick based PID controller as well as the GA based controller.

Since the fuzzy controller is a model free approach, it provides a constant performance, even though the non-linearity and uncertainty may change. Furthermore, it is possible to run the controller on different speeds when the controller runs down the nut on the bolt and performs the tightening. In addition, the error recognition is directly integrated into the fuzzy logic which allows speeding down the controller when torque and angle levels become critical.

The four stages fuzzy controller is based on the torque angle tightening technique. The PI and PID controllers are solely based on the angle tightening technique. They can be combined to a single controller which will control both the torque and the angle, but this means that two PI or PID controllers need to be introduced with an individual numerical model for torque as well as for the angle.

Summarizing, the best control strategy is the fuzzy control strategy with regards to error detection and performance. Even though the accuracy significantly increases with the fine-tuned GA PI Control strategy, this occurs only after the training process has been completed. Therefore, the first hub with three blades (each bearing has up to 128 bolts, so 384 bolts in total) will not have the tightening completed on a constant level. Only when the GA has improved the gains of the PI controller, the final tightening results achieves a high accuracy as shown in Figure 58. The fuzzy controller includes expert knowledge about the tightening process using membership functions and linguistic rules rather than a numerical bolt model. The integrated expert knowledge allows to provide a constant control result – the final torque and angle levels will always be reached. Therefore, the four-stage fuzzy controller is the best option for the wind turbine hub assembly process.

5.2 Error detection capabilities comparison

The error detection capabilities have been compared for both the four stages fuzzy control strategy as well as for the genetic algorithm based self-tuning PI controller.

Both strategies performed equally as far as error recognition was concerned. The fuzzy controller however has the extra advantage of running on different speeds which reduces the kinetic energy when an error occurs. Furthermore, due to the linguistic rules the tool slows down as soon as the control values indicate that an error scenario may be arising decreasing the kinetic energy in the tool and the bolt system even further.

The logic based error detector, which has been implemented for the self-tuning PI control strategy showed 100% performance on all defined errors. However, it is a program which runs parallel to the PI controller. The controller output is depending only on the input error (e.g. the error in the tightening angle) and the output is therefore on its maximum. The tightening tool runs then on maximum speed and if an error occurs in this situation (e.g. when the nut is placed on top of the bolt) the maximum kinetic energy will be applied to the threads of the bolt and the nut. It did not cause any damage in the used set up but if a different bolt material is used in a different assembly application it may do. Therefore, summarizing, the error detector in the four stages fuzzy controller is the preferred solution.

Table 10 compares the error detection capabilities of each control strategy.

ERROR SCENARIOS	DETECTION [%] FUZZY/GA	DETECTION STAGE [%] (FUZZY CONTROLLER)				TORQUE LEVEL (GA)	ANGLE LEVEL (GA)
		S1	S2	S3	S 4		
S2: Misalignment	100 / 100	100	-	-	-	>10Nm	<15°
S3: Jamming	100 / 100	10	90	-	-	<5Nm	>50°
S4: 2 washers	100 / -	-	-	33	67	-	-
S5: Nut missing (M24 nut)	100 / -	-	-	100	-	-	-
S6: Small bolt, big nut	100 / 100	-	-	100	-	>10Nm	0°-47°

Table 10: Comparison of all tightening trials

The scenarios S4 and S5 are not included in the logic based error detector. This is due to the fact that it is using an angle based tightening technique where the nut runs down to the flange and the washer. As soon as it reaches the flange, there is a peak in the torque level which defines the zero angle position. Therefore, two washers or a missing nut cannot be detected. However, if a timer is used and the times are estimated experimentally, this can be integrated into the logic based error detector.

The four-stage fuzzy controller defines its zero position when the nut is placed on top of the bolt. Based on the angle, the scenario S4 and S5 can be detected.

CHAPTER 6

6. Chapter 6: Outlook: Assembly of the wind turbine hub and the bearing

Outline: Overview flexible wind turbine hub assembly

This chapter extends the previous chapters to a wider concept to assemble the wind turbine hub and the bearings.

The wind turbine is made out of several components and the three main components have been analysed to be included in a reconfigurable assembly strategy. The assembly for each item has been described and stored in a database.

The assembly strategy is based on an agent which detects changes in the database and starts a search algorithm, based on Partial Order Planning (POP).

The POP will combine the assembly steps by its pre- and post-conditions (causal links) to a complete assembly strategy.

Summarizing, the strategy has been simulated in an automation environment. The results have been recorded and presented.

6.1 Chapter structure

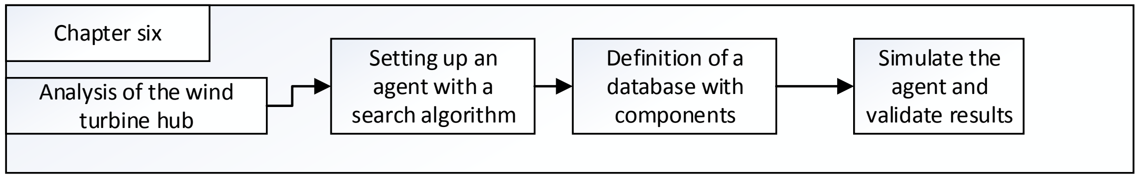


Figure 59: Chapter five structure

6.2 Outlook on agent based assembly of wind turbine hubs

This chapter introduces agents and planning algorithms to be used in the assembly of the wind turbine hub. Wind turbines are constantly improved and their parts are upgraded on a frequent base as research and development continues [9]. In manual assembly lines, upgraded parts can be fairly easily integrated without the assembly line shutting down for this purpose. However, an automated assembly needs to be shut down and reprogrammed in order to integrate a new part.

An agent is an in depended software program which acts autonomously to achieve goals. It can interact with other agents and with its environment and respond to changes and other actions [66]. It is gaining its information in real time and is connected with sensors and actuators in the manufacturing environment to interact with physical processes [67] [68].

Agents can be implemented using fuzzy logic for monitoring and diagnosis [69] and also for decision making processes [70] - [71].

Since the assembly process contains several sub routines for each part of the turbine, an agent based automation concept can be introduced to provide the flexibility and adaptability for integrating changes and new parts. This concept can be combined with search algorithms, such as the partial order planning search algorithm which provide adaptability for changing and integrating a new assembly strategy of an item.

Another way of implementing an agent for the wind turbine hub assembly is using a search algorithm, such as the Partial Order Planning algorithm (POP) [72]. The POP algorithm can be implemented within an agent. The agent starts the search algorithm in order to find a suitable assembly strategy based on the parts stored in the systems database.

The wind turbine contains several components and for each component assembly specifications need to be defined. The assembly specifications contain information on how the item should be assembled as well as pre-requirements and post-requirements which define what needs to be completed before the actual assembly (pre-state) and what can be done after the assembly of the part is completed (post-state). To design the strategy, the components of the wind turbine hub need to be analysed and their assembly strategy need to be defined. All these parts are stored in a system database.

The hub contains three main parts, which is the hub body, the pitch system and the bearings. For this strategy it is assumed that the pitch system is already installed in the wind turbine hub.

6.3 Structure for the agent based assembly framework

The framework for the agent for the hub assembly has been planned as shown in Figure 60. It is an individual agent which can be linked to other agents in the assembly line (such as an agent for the assembly of the nacelle). Furthermore, the four-stage fuzzy controller and the self-tuning genetic algorithm can be integrated into the framework.

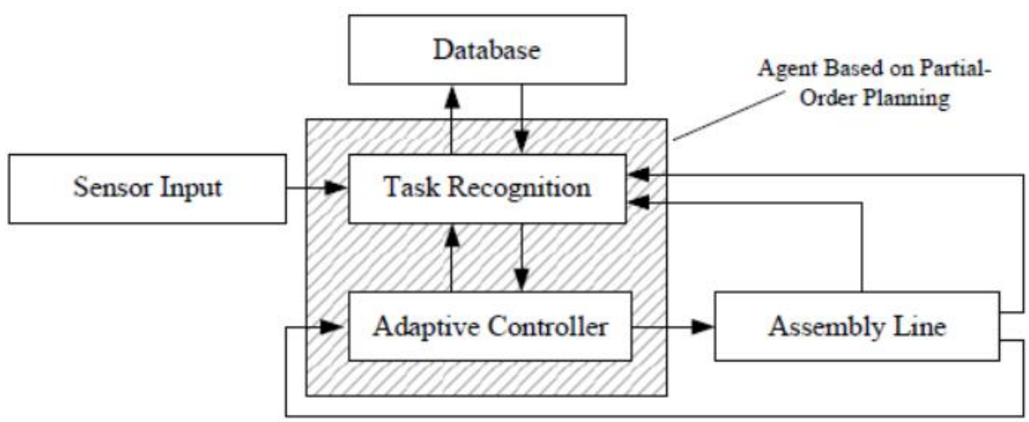


Figure 60: Reconfigurable assembly strategy using an agent

The shown strategy contains 5 main areas, the sensor input, the task recognition, the database and the adaptive controller as well as the assembly line.

Task Recognition:

The task recognition block runs a search algorithm (POP) which is looking at pre- and post-conditions of the assembly actions stored in the database. The algorithm will start to find the first action (which is the one without a pre-condition) and continues by searching for the next action where the pre-condition matches the post-condition of the first action. It will do that until it reaches the final action (which is the one without a post-condition). The task recognition block will remain active and constantly look at the database for an update. As soon as the database is updated the search algorithm will be executed again to detect a changed assembly instruction (action) for a changed item. It will then update the assembly strategy and download the strategy into the adaptive controller, which is directly linked to the assembly line. It should be noticed that while the search algorithm is running the actual assembly line is running as well and does not need to shut down for the update. The search algorithm can therefore work in parallel using several PLC cycles.

Adaptive Controller:

The adaptive controller is directly linked to the assembly line. It performs the actual assembly procedure, which means that it executes the assembly actions on the order the task recognition block has defined them and downloaded them into the adaptive controller block. It is noted that the update of the adaptive controller block requires only one PLC cycle; after which the new assembly strategy is immediately online. It just takes a few PLC cycles to complete the update process of the adaptive controller.

Assembly Line:

The assembly line block represents the physical factory assembly line connecting the actuators with the adaptive controller.

Database:

The database contains all actions, pre- and post-conditions, which are required for the assembly of the wind turbine hub. The database needs to contain the most recent part description plus an assembly instruction set (action) including its pre- and post-conditions which describe the required (pre-condition) assembly stage for the instruction set to be completed as well as a description of the situation after completion of the instruction set (post-condition).

The database is constantly monitored by the agent running the search algorithm (task recognition). The assembly strategy is at first created when the system is started and then it is continually updated when a new item is introduced to the database or when an item is modified or amended in its pre- or post-conditions or when the assembly instruction set is changed.

Sensor Input:

The agent based system requires real time information from the assembly line. The sensor input block symbolizes all the sensors in the system whereas the assembly line block (which actually includes the sensors) describes the actuators and the physical structure of the assembly line.

6.4 Optimization using Partial Order Planning

The Partial-Order Planning algorithm [72] is of special interest in this research case since it can be used and extended for automation approaches in the assembly industry.

This algorithm is chosen because the strategy which it returns is always correct and complete. If the algorithm cannot return a strategy, the assembly actions and the pre- as well as the post-conditions need to be revised as there is either an incorrect or missing action or condition.

The POP algorithm consists of three main parts, which are a set of ordering constraints describing the pre and post conditions of each step. The following actions can be defined for assembly process:

1. Hub placement at the assembly station
2. Hub preparation (Bolt hole check, installation)
3. Bearing installation
4. Bolt nut tightening

6.5 Conditions for Assembly Actions

Connections between the actions are given by pre and post conditions, which are defined. For action 1, the hub needs to be available, in a correct size and ready to be placed in the assembly station. Action 2 includes two internal actions, where the hub is prepared. The overall action requires the hub to be ready in the assembly line which is a requirement for the 'bolt hole check' action. This is followed by the bolt installation, where the bolts are placed in the turbine hub. Post-conditions are defined by fulfilling the pre-conditions. Bearing installation (action 3) requires that the bolts are installed. Post-conditions are set by an installed bearing and the bolts are ready for nut tightening. Action 4 is the final action for completing the assembly of hub, where the nuts are tightened on the bolts. The conditions for executing this action expect an installed bearing as well as installed nuts. These actions can be summarized using first order logic.

Actions, requirement and return values	**Assembly**
Action 1	Place Hub(Astation, Warehouse):
Require	¬At(Astation, Hub) ^ At(Hub, Warehouse)
Return	At(Astation, Hub) ^¬At(Hub, Warehouse)
Action 2	Prep Hub(astation, Hub):
Require	At(Astation, Hub)
Return	At(Hub, holes) ^ At(Hub, bolts)
Action 3	Bearing Inst(bearing, Hub):
Require	At(bolts, Hub) ^ At(Astation, Hub)
Return	At(Bearing, Hub) ^ ¬At(Bearing, Nuts)
Action 4	Nut Tighten(Nuts, Hub):
Require	At(Hub, Bearing) ^ At(Hub, Bolts)
Return	At(Nuts, Tighten)

Table 11: Action table

6.6 Partial Order Planning and Assembly Strategy

The search algorithm will search for the assembly strategy and when the strategy is found it looks like the strategy as shown in Figure 61. This defined step chain can be stored in the database and transferred to the controller. Any changes can be introduced and the chain order redesigned. In the turbine industry, there are many different turbine hubs with various sizes of bolts, bearings etc. As wind turbines are constantly developed, a change can be introduced by using this algorithm. In fact, the advantage of using an intelligent agent is that it can check whether there is a new or updated action in the database which is linked to the assembly process. It can then automatically re-design the assembly chain and, since the POP algorithm is always correct and complete, it can automatically generate a new step chain which is then executed in the next PLC cycle. So, neither re-programming nor a shutdown of the machinery is required.

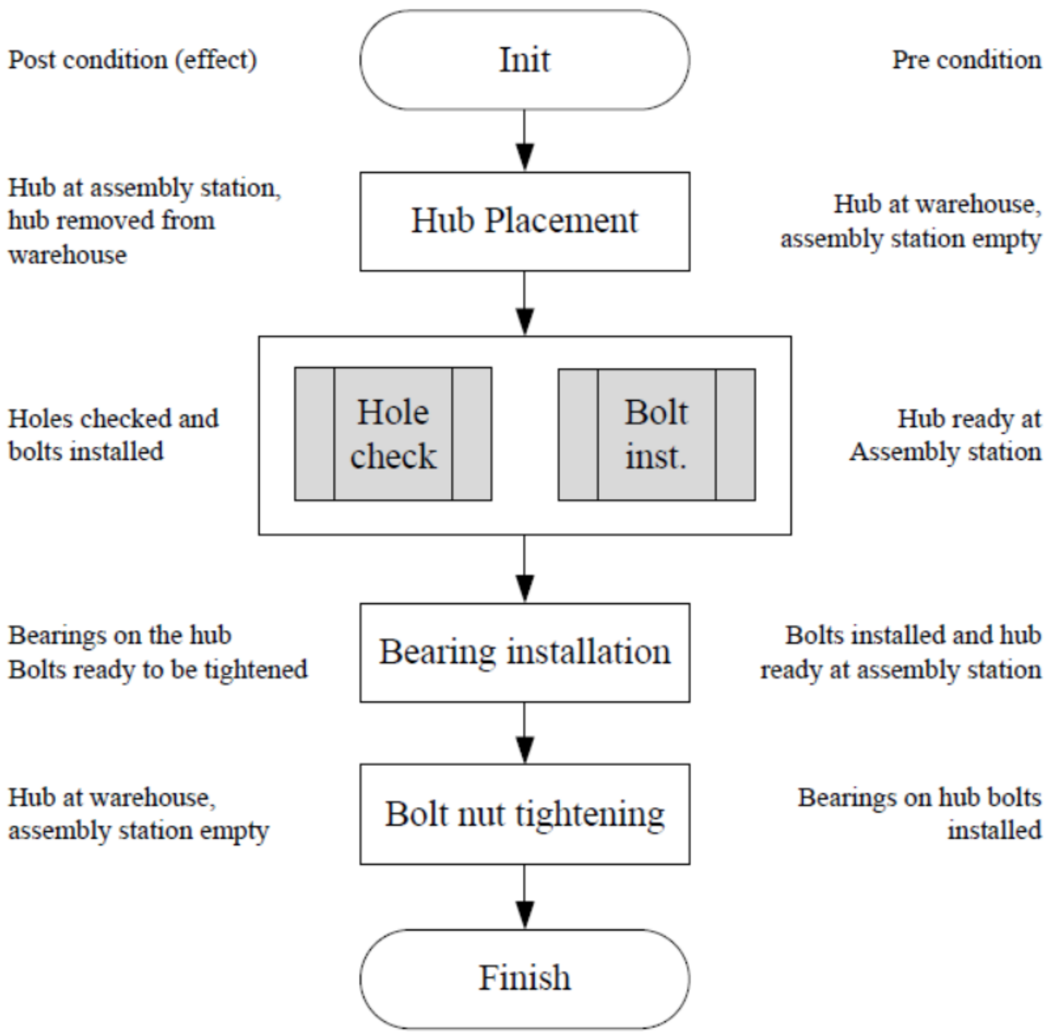


Figure 61: Adaptive assembly strategy

The presented industrial control is implemented on a standard industrial controller programming tool called TwinCAT 3 provided by Beckhoff Automation GmbH. TwinCAT is an automation software environment. It integrates into Visual Studio 2010 and builds a bridge between classical automation and computer science. The system architecture is illustrated in Figure 62. In order to set up the controller based on Partial-Order Planning, MATLAB is chosen. Standard PLCs run normally based on a sequential programming language, such as Structured Text (ST), which is an intuitive programming language. MathWorks introduced recently the PLC coder, which allows code conversion from Simulink models to ST code, which is limited basic Simulink functions as well as Stateflow models. Beckhoff TwinCAT 3 uses integrated C++ code from MATLAB Windows real-time target. The agent is defined as a Simulink model, converted into C++ code, and then integrated into a PLC cycle. The schematic structure of TwinCAT 3 is shown in Figure 62.

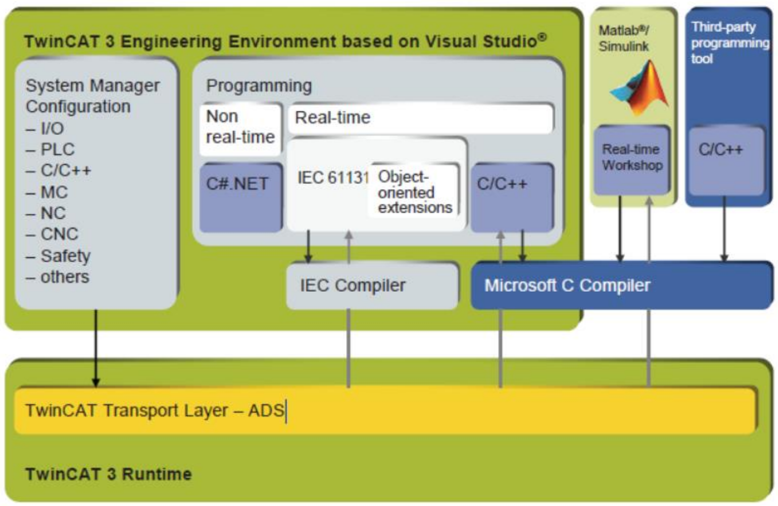


Figure 62: Beckhoff TwinCAT 3 Automation System structure [58]

The simulation results shown in Figure 63 show the switching moments between the database update, the search algorithm as well as the adaptive controller. The adaptive controller also represents the assembly line.

As it can be seen, when the simulation starts, the agent waits until the update process of the database has been completed. Once this is done it starts the search algorithm (partial order planning) in order to derive an assembly strategy. At the same time, once a strategy has been derived it is downloaded to the adaptive controller and the assembly line starts. The time line shows PLC cycles; the cycle time for this simulation has been set to 1ms. This shows that the adaptive controller can start the assembly line after 28ms. However, this time is depending on the completion of the search algorithm and with further increasing complexity it may take more PLC cycles and also more time (it also depends on the performance of the used industrial control hardware, if the system is slower, it may take longer).

The next simulated scenario has been extended by an update scenario, where an item in the database is either updated or replaced with a different item and therefore a new assembly action strategy is introduced. The update process is started at 29 PLC cycles. The agent running the search algorithm will start the search algorithm again until the database update is completed. This is continued by an update of the assembly strategy from the task recognition block. The new strategy is then downloaded into the adaptive controller which is restarted as soon as the download is completed.

In case of another update the process is repeated. Therefore, it is not required to shut down the assembly line for amendments of the assembly strategy, only the part description needs to be renewed by the R&D Department and the database needs to be updated. This keeps the downtime of the assembly line to a minimum.

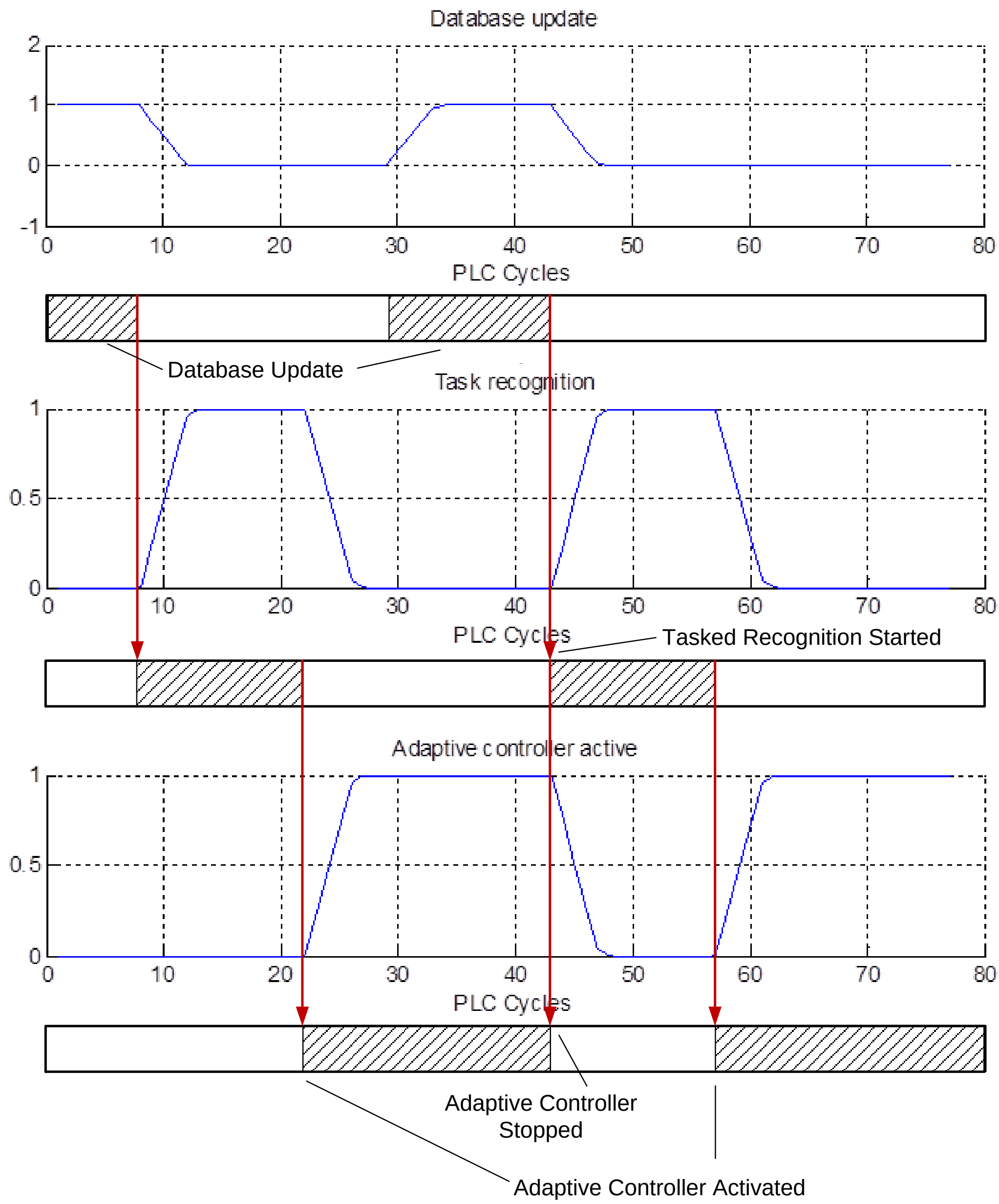


Figure 63: Simulation of the POP Algorithm

6.6 Summary

This chapter introduces an agent based concept for the automated assembly of wind turbine hubs. The assembly strategy has been generalized to the main components for proofing the concept.

The assembly has been analysed and the hub assembly has been subdivided into a set of instructions describing the assembly of each part of the wind turbine hub. For the overall assembly, an agent-based concept has been suggested where individual agents ensure the assembly for each group of components (e.g. the wind turbine hub, the nacelle etc.)

The agent-based concept contains 5 main components, namely 'sensor input', 'adaptive controller', 'task recognition', 'assembly line', 'database'.

The R&D Department will need to provide a description for each item which is required within the individual assembly group (in this example the wind turbine hub). The item descriptions as well as the pre- and post-conditions are then uploaded into the database. An agent will look at the individual assembly group and will create an assembly strategy based on the pre- and post-conditions to derive a suitable strategy. For this application, the partial order planning search algorithm has been chosen, since it is capable to find an assembly strategy and ensures completeness and correctness. The algorithm has been tested in a simulation environment based on an industrial real time environment using MATLAB/Simulink and Beckhoff TwinCAT 3. Several cases have been defined and simulated, starting from the first start-up of the system, where the database is empty and will be updated. The agent waits until the update has been completed and runs the search algorithm. Once the search has been completed the defined assembly strategy is downloaded into the controller (adaptive controller) which is then linked to the assembly line.

The system then starts the assembly process and the agent constantly checks the database for any updates. Once it recognizes an update, it starts the search algorithm again and derives a new assembly strategy. The adaptive controller will then restart the assembly process with the new updated strategy.

The simulation results proved that the concept works, that updates can be completed within a few PLC cycles and that the update of the assembly line can be done in real time.

CHAPTER 7

7. Chapter 7: Summary & Conclusions

Outline

The wind turbine industry is a growing industry where automation is currently being introduced to reduce the costs of the assembly.

In this thesis, the wind turbine hub assembly process has been investigated with a view to propose new strategies for process automation. This PhD study has focused on two main elements of the overall wind turbine manufacturing process: (a) the bolt tightening process of the wind turbine bearings and (b) the wind turbine hub assembly.

The bolt tightening process is one of the core processes and requires to be completed with high accuracy to ensure that each nut/bolt connection is competed with the required clamping force. If the clamping force varies, a local bolt connection may break and cause wind turbine failure.

Control algorithms for the tightening have been designed employing model-based and model-free approaches. The model-free approach is based on fuzzy control incorporating expert knowledge of the tightening process. The model based approach makes use of a numerical model of the bolt representing the physical behaviour of the nut/bolt system.

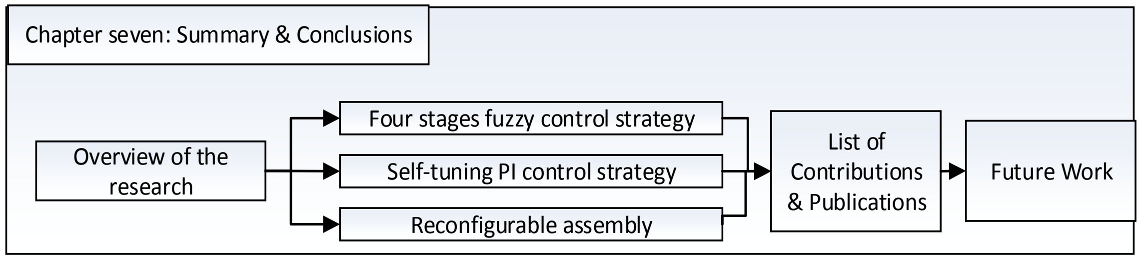


Figure 64: Chapter 6 structure

7.1 Summary and research background

The wind turbine assembly contains several steps; the final installation is completed in the field where all components are mounted together.

The pre-assembly is done in the factory, where the wind turbine components are pre-assembled. The assembly contains several individual steps which can be grouped.

In the next step, an agent based concept is introduced where agents are linked to each assembly group. The agents will then interact to assemble the wind turbine. Each assembly station in the factory may have an individual agent which interacts with other agents required in the particular assembly chain.

For the concept an agent has been described for the assembly of the wind turbine hub. It is a reconfigurable concept where components can be updated in a database the agent runs a search algorithm based on partial order planning to find the correct assembly strategy.

In this case, the R&D department can develop the parts for the wind turbine and describe how they need to be assembled. Furthermore, pre-and post-conditions need to be defined which the partial order planning algorithm will then link together by creating causal links between the matching pre- and post-condition. This allows to integrate changes to the assembly line quickly and without the need to shot down the assembly line, reprogram the automation system (PLC) and restart the assembly line.

The other part of the research focusses on the bolt tightening process. The bolt tightening process is a highly non-linear process with uncertainties making classical control approaches difficult. Furthermore, it is essential that the tightening is completed accurately and that the correct bolt and nut combination is used with the correct washer.

For the research the focus has been on a model-free and a model based strategy. The model-free strategy is using fuzzy control to address the non-linearity and uncertainty of the system by including expert knowledge of the process. Furthermore, the error recognition functionality is also included by using linguistic rules and membership functions to directly address several error types allowing to classify them.

In addition, the tightening process has been divided into four different stages, which is the placement of the nut on top of the bolt, the alignment of the nut, the run down until the nut reaches the flange and the final tightening process.

The model-based strategy is based on a state-space model which has been derived using the MATLAB system identification toolbox. It has the same inputs and outputs as the bolt system including the tightening tool. The model however is not able to represent the non-linearity and uncertainty of the system meaning that it only approximates the system. The gains which have been derived using this model are able to achieve the control target but can be further improved with the online learning Genetic Algorithm.

The performance is rather low, since the control strategy based on PI control is now facing the non-linearity and uncertainty of the physical system. However, it is still possible to reach the control target.

For further performance improvement, a Genetic Algorithm has been used to learn the non-linearity and uncertainty of the system and further fine-tunes the gains of the controller while it was running in real time in the assembly environment. Around the pre-estimated gains boundaries have been defined in a way, that the control target can be reached and the controller is improved within the gain boundaries until the optimal gains have been derived for the particular set of bolts. Furthermore, when a new set of bolts is introduced to the system, the genetic algorithm will adapt the new system attributes and estimate further optimize gains.

On top of the self-tuning PI strategy, an error recognition block has been added. This is based on a logic which monitors the torque and angle values. Predefined torque and angle levels will set a flip-flop in the Simulink model, which stops the control strategy.

Comparing both control strategies, it could be shown that the final control result for the final fine-tuned self-tuning strategy the control result is better than the four stages control strategy based on fuzzy control. However, this only applies for the final result, the average result is performing not as well as the four stages fuzzy control strategy. This means, with regards to robustness of the controller, the four stages controller performs better as the mean value is closer to the target value and the standard deviation is lower compared to the PID control strategy.

Furthermore, due to the four stages including expert knowledge the non-linearity as well as the uncertainties by using expert knowledge. Each stage addresses the particular non-linearity and uncertainty of the particular stage. Error recognition is also implemented on each stage allowing a classification. This PhD study contributed to an EU project called COSMOS. The project is linked to the wind turbine manufacturing industry.

7.2 Suitable control architectures and test bench design

In general, for the preparation of the research various suitable automation systems have been looked at and requirements have been defined in order to run the control strategies fast enough (cycle time) as well as integrating them into the factory environment.

A professional automation system (Beckhoff TwinCAT 3 [58]) and bolt tightening system (DSM Messtechnik MDW 140) has been selected and used for the automation application, since it allows integration in a factory environment as well as easy interaction with other actuators in the assembly line, e.g. robots.

The other essential part for preparing the research was the design of a torque reaction system to prevent backwards torque into the robot arm (which can be extended into a larger system to be used on a wind turbine hub, in the research only a test bench has been used as shown in Appendix B).

Another test bench has been designed for constant testing of a control strategy, the technical details are also introduced in Appendix B. This bench keeps the tightening tool and the bolt in a position and allows several automated experiments without the need of the robot arm, a clamping force sensor can also be integrated between the flange and the nut to measure the final clamping force of the bolt connection.

Furthermore, in case an experiment goes wrong, the tool can apply maximum torque and there is no risk of any physical damage.

7.3 Summary of the control designs for the assembly of wind turbine hub bearings

The bolt tightening process for the hub is a non-linear process including uncertainties. Two control strategies for the bolt tightening process have been derived for the assembly of the wind turbine hub bearings.

7.3.1 Design of a four stages fuzzy control strategy

The bolt tightening process has been analysed and divided into four different stages, namely, the nut placement, the alignment, the run down phase and the final process, where the tightening process is completed.

An in depth analysis has been carried out and expert knowledge has been developed to design a novel fuzzy control strategy. The novel strategy can address the non-linearity and uncertainty of the tightening process [16]. The expert knowledge has been defined using membership functions as well as linguistic rules.

Another essential novel achievement is the error recognition. This is implemented based on linguistic rules and membership functions based on the expert analysis which sets an error flag. Furthermore, the controller runs on different speeds to address specific problematic areas, such as during alignment to avoid mechanical damage of the bolt. Together with the information of the stage where the error occurred it is possible to classify the error. Also, with the membership functions, the tools slow down before an error scenario arises to minimize any possible physical damage. There is currently no other tightening algorithm which combines both the accurate fuzzy tightening control strategy as well as the error recognition capabilities.

7.3.2 Design of a self-tuning PI controller using a genetic algorithm

Another novel developed strategy is based on a PI controller to introduce a new concept which can learn the non-lineareties and uncertainty. In this concept a PI controller has been derived based on a numerical model which represents the general system response of the bolt system. As a start, the gains have been derived based on this model so that the control target can be reached.

The controller can be then further fine-tuned by using an online Genetic Algorithm to learn the system attributes of the bolt system. This integrates the non-linearity and uncertainty of a particular set of bolt.

Furthermore, error recognition is also essential on this control strategy. This has been implemented with a logic based algorithm which runs in parallel to the PI control strategy. When an error occurs the logic sets an error flag which stops the controller.

7.4 Design of a self-reconfigurable strategy for wind turbine hub assembly

As a final step, a novel extension of the system has been suggested to integrate the tightening algorithms into a novel wider concept using agents. For this application the wind turbine hubs have been divided into its main components, which are the pitch system, the hub body and the bearings.

The assembly strategy (assembly action) for each component is described by the R&D Department and it is stored in a database. Each action has pre- and post-conditions (causal links) which describe what needs to be done first before the action can be applied and what can be done after. A search algorithm based on Partial Order Planning (POP) is used in a simulation environment to show the performance of the system.

Parts of the assembly can be replaced in the database; the POP is started by an agent which monitors the database. Once a replacement has been detected, a new assembly strategy is derived and downloaded into the controller.

7.5 Summary of Contributions

A summary of the major contributions made by this research to the field of wind turbine manufacturing is shown below. The research results have shown that the developed new strategies can make the manufacturing process safer, they prove to reach a better precision, provide high level error detection to reduce physical damages and downtime of the assembly line and provide a better overall efficiency of the assembly process. The research can also be applied to a wider field, such as the aeroplane industry where bolt connections are used to set up the internal structure of the plane.

<u>Proposal of a novel bolt tightening technique based on a model free fuzzy controller</u>.

The study of the bolt tightening assembly process leads to four different stages based on the steps for the assembly and tightening process, which is the placement of the nut on the bolt, the first few turns to install the nut on the bolt followed by the rundown of the nut and the actual tightening process. A similar process can be found in a screw insertion process as introduced in [73], where the insertion of a screw is described and also different stages are identified.

For model-free based fuzzy control, the bolt tightening algorithm has been based on the torque/angle technique [16], where the nut is run to a specific angle as well as a specific torque level. For the fuzzy control strategy, the inputs of the control system are torque and angle and the output of the system is a speed signal which is connected to the bolt tightening tool (based on an analog voltage signal). The tightening process uses both input signals as well as the output signal have been described using membership functions and have been linked using linguistic rules. The error scenarios have also been described using membership functions and linguistic rules. The controller provides an error output, which shows when an error occurs. Together with the information about the tightening stage when the error happened the error can be classified (e.g. stage 1 error can be a misalignment of the nut on the bolt).

Furthermore, since the tightening process has been divided into four stages it also allows to run the tightening tool on different speeds to prevent potential damage in each stage by allowing the tool to respond to the error quicker based on the low tightening speed. The tool runs slower in stage 1 for example, where the nut is placed and a misalignment needs to be detected. In an error scenario (misaligned nut) the nut will get jammed. If the tool runs slow in this event, the kinetic energy in the bolt tightening system is low and when the tool is switched off there will be less kinetic energy left in the system avoiding overshoots if the tightening is suddenly terminated.

The implementation has been done using an industrial automation system. The novel control algorithms can be therefore directly ported into factory environments.

Proposal of a novel model-based control strategy and online learning algorithm:

As preparation, the bolt system has been analysed in depth and experimental data has been recorded on how the bolt system in combination with the tightening tool responds. A sine wave has been set up and applied to the tightening tool. The response in torque and angle has been recorded and the result has been applied to the MATLAB System identification toolbox and a model has been derived. It turned out that the model is not able to represent the non-linearity and uncertainty (caused by variation in friction, temperature, material) in the system. The model has been used to estimate the gains of a PI controller and the controller has been tested on the bolt system. As a result, it turned out that the performance can still be further improved.

For handling the non-linearity and uncertainties, the PI control algorithm has been linked to a Genetic Algorithm (GA), which processes the system response after each tightening process and runs after each completion of a tightening process. The combination of both the GA and the PI controller allows to find the best Gains for the PI controller.

Within experimental predefined boundaries around the control gains (a range where the GA can set new gain values) the P and I gains will be optimized for the system and fine-tuned until the response is improved with regards to time and accuracy. In this way, also when a new set of bolts is introduced to the system the GA will learn its attributes and adjust the PI values to provide an improved performance as the new set of bolts may require new control gains.

Finally an error detection capability are included, which is essential in this control strategy. For the implementation, a logic based error detector has been developed with similar error types when compared to the previous proposal based on fuzzy control. The logic based algorithm runs in parallel to the controller and can stop the process when an error occurs. It will then return an error with the torque and angle values so that the error can be classified.

The experiments and implementation has also been done on an industrial standard control system.

<u>The final contribution is a novel approach using an agent based framework allowing the wind turbine hub assembly can be automated with a high degree of flexibility:</u>

The wind turbine is constantly further improved and parts of it are modified [9]. In an automated system. The assembly line needs to be shut down and the automation strategy needs to be reprogrammed in order to include the new or modified item in the assembly. This proposal introduces an on how a change over in assembly can be done in an adaptive manner without the need to shut down the assembly line when a new item is introduced.

The wind turbine hub has been analysed and generalized to three main components to design a system which can derive an assembly strategy automatically by itself. The wind turbine hub has been broken down into three main parts, which is the hub body, the bearings and the bolts. All the components (different hub sizes, bolts and bearings) have been stored in a database with their assembly specifications including a description on how they need to be assembled (defined as an action), what needs to be done before (defined as pre-condition) and what can be done after (defined as post-condition).

The pre- and post-conditions can be linked by a search algorithm, as all pre- and post-conditions are set in a way that they can be linked (defined as a causal link). For this application, a search algorithm has been chosen which is always complete and correct, the partial-order planning search algorithm (POP). If it is not successful, it means that there is a problem in the definition of pre-and post-conditions. Even though other search algorithms may also work the simulation results proof a good performance in this application.

Furthermore, the search algorithm is implemented as an agent. The agent will look at the database and checks for changes. Once a change is detected it will start the search algorithm which derives and updates the assembly strategy. The concept also outlines that agents can be used for each component of the wind turbine, such as the nacelle. However, as it is only an outlook, the concept only for the hub has been tested and implemented.

7.6 Future Work

The introduced strategies have shown a good overall performance and is on a level that it can be integrated into a factory environment.

The following suggestions are made for further improvements.

7.6.1 Extending the four-stages fuzzy control strategy

The four stages fuzzy controller can be extended to have several rules and membership functions to include bolt parameters for the assembly of the wind turbine hub to cover all the wind turbines which are assembled at that factory.

Another decision making algorithm based on fuzzy control can be implemented which switches between the fuzzy control strategies for each bolt size.

Furthermore, the error recognition capability should also be amended so that it can handle various sizes of bolts used in the assembly application.

Moreover, even though the performance of the control strategy is on a good level and it shows a robust behaviour, it may be combined with the introduced genetic algorithm to fine-tune the Gaussian membership functions for a better result. This can be done by defining boundaries around the predefined values of the membership functions.

7.6.2 GA fine-tuned PI control strategy

The GA fine-tuned PI Control strategy for the bolt tightening process is based on the angle tightening technique. Even though this strategy provides an accurate tightening technique it takes several tightening processes until the GA has learned the non-lineareties and uncertainties of the bolt system. Especially the starting point of the threads is varying on each bolt (though they are from the same set of bolts) which is depending on the installation.

The strategy can be improved by using the torque/angle tightening technique rather than only the angle tightening technique. This would increase the accuracy, but may require two PI controllers which control the torque level as well as the angle level. Two PI controller which are combined with a GA will also increase the computational requirements of the strategy.

In addition, it may be more suitable to replace the PI Control strategy with a different strategy, such as a state-space controller. The gains can also be fine-tuned using a Genetic Algorithm to learn the system parameters to integrate the non-linearity and uncertainties.

Furthermore, the four stages fuzzy controller as well as the GA fine-tuned PI control strategy can be integrated into the system.

7.7 List of Author Publication

The research results have been published in the following journal and conference proceedings:

Journal Publications:

[1] **C. Deters**, H. K. Lam, E. L. Secco, H. A. Wuerdemann, K. Althoefer, "Model-Based Self-Tuning PI Control of Bolt-Nut Tightening for Wind Turbine Bearing Assembly", IEEE Transactions on Industrial Informatics (under review)

[2] **C. Deters**, H. K. Lam, E. L. Secco, H. A. Wurdemann, K. Althoefer, "Accurate Bolt Tightening using Model-Free Fuzzy Control for Wind Turbine Hub Bearing Assembly", IEEE Transactions on Control Systems Technology

[3] H.K. Lam, Hongyi Li, **C. Deters**, E.L. Secco , H. Wuerdemann and K. Althoefer, "Control Design for Interval Type-2 Fuzzy Systems under Imperfect Premise Matching", IEEE Transactions on Industrial Electronics 2013

Conference Publications:

[4] **C. Deters**, E.L. Secco, H.A. Wuerdemann, H.K. Lam, Lakmal Seneviratne, K. Althoefer, "Model-free Fuzzy Tightening Control for Bolt/Nut Joint Connections of Wind Turbine Hubs", IEEE International Conference on Robotics and Automation, 2013

[5] **C. Deters**, H.A. Wurdemann, Prof L.D. Seneviratne, and Prof K. Althoefer, "Reconfigurable Assembly Approach for Wind Turbines using Multiple Intelligent Agents", ASME/IEEE International Conference on Reconfigurable Mechanisms and Robots, Tianjin, July 2012

[6] Mohammad Narimani, H.K. Lam, K. Althoefer, R. Dilmaghani, Charles Wolfe and **C. Deters**, "An Approach for Stability Analysis of Polynomial Fuzzy Model-Based Control Systems", IEEE International Conference on Fuzzy Systems, 2011

APPENDIX A

8. Appendix A: Test Bench & Simulink Models

Appendix A provides an overview of the developed test bench, the four stages fuzzy control strategy as well as the PI GA Algorithm. Only the structure is described, a copy of the complete program can be found on the attached DVD of this thesis.

8.1 Appendix A Structure

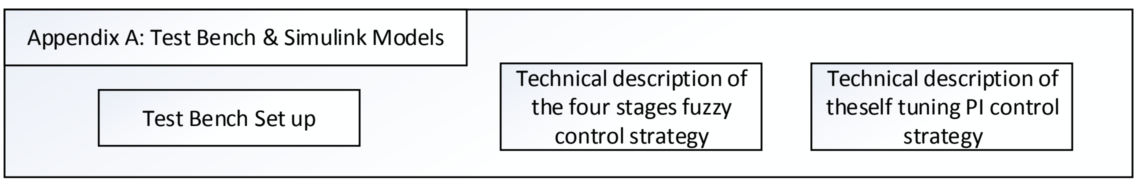


Figure 65: Appendix A Structure

8.2 Test Bench setup

The application of assembling wind turbines is an industrial application and requires to be set up in a way that the results of the control design can be directly transferred to the real physical application in the assembly line. The setup has 2 parts which need to be considered; on the one side it’s the platform for running the control algorithm (software), on the other side (hardware) it is the tool for tightening the bolt.

8.2.1 Test bench requirements for the hardware (tightening tool set up)

One essential part of the wind turbine hub assembly is the bolt tightening process. To do this, a tightening tool is required and it should have the following specifications:

- Industrial standard
- Contains a strain-gage torque sensor for accurate torque measurements
- Contains a encoder to measure the angle
- Can be directly connected to a PLC set – up using analogue signals
- Can be downscaled: 3000Nm is the maximum tightening force, however, for demonstrating a lower value can also be considered
- Can be integrated to the Fanuc Robot arm
- Able to be integrated into a clamp system to prevent backward torque into the robot arm

<u>Using a servomotor with an encoder</u>

In order to tighten the bolts of the wind turbine hub a tightening tool is required. A simple way of realizing this would be using a servo motor with a controllable power transducer. The torque can be controlled by measuring the current of the motor and therefore, the angle as well as the torque can be used. Furthermore, this solution allows a fairly straight set up and minimizes also the costs. However, the project targets to use a classical industrial tightening tool. Hence the industrial tools provide already certifications it is in long term the cheaper and easier solution to set up.

<u>Using an analogue tightening tool type DSM BL 57 140 MDW</u>

Another look has been taken to industrial tools. There are a few specialized manufactures and a detailed look has been taken to a tool produced by DSM called DSM BL 57 140 MDW. The tool is shown in Figure 66.

Figure 66: Tightening Tool DSM BL 57 140 MDW

This tool has already all certifications to be used in a factory environment and is therefore suitable to be used for this application. The chosen tool has the following specifications:

- Provides CE certification and EMV compatibility
- Uses analogue input signals between ±10V
- Provides an integrated encoder for measuring the angle (accuracy 1%)
- Provides a strain-gage sensor for measuring the torque (accuracy 1%)
- Can be integrated into the FANUC robot set up
- Is able to carry a clamp to avoid any backward torque into the robot arm
- The rotational speed can be set using an analogue speed signal

The tightening tool has been selected and set up in 2 ways; one is a tightening bench to easily run the tool on an individual bolt without using the robot arm and the other set up is based on a torque reaction system to avoid any backward torque. The bolt bench is representing the wind turbine hub. For the physical application in the factory the clamp system needs to have the correct size of the bearing so that it can create a closed torque reaction system with the bearing to prevent backward torque.

This is shown in Figure 67.

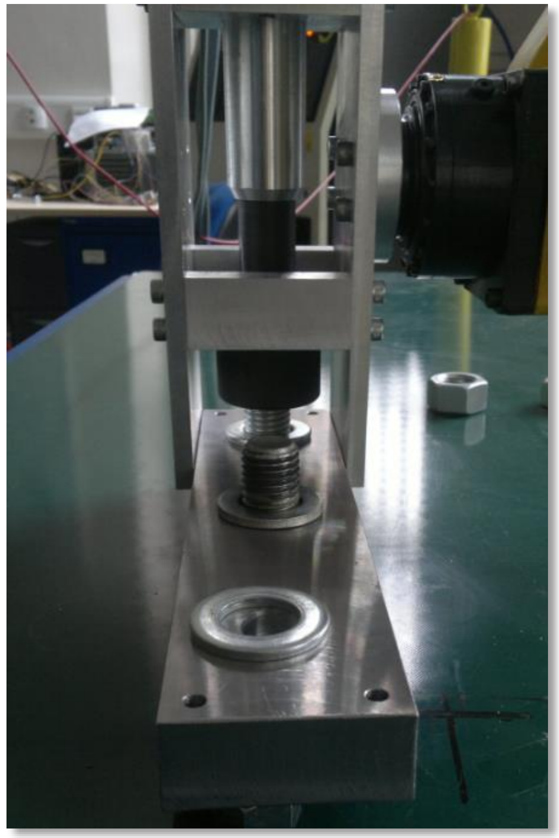


Figure 67: Torque reaction system to prevent backward torque in the robot arm

Continuously, the tightening tool requires a standard connection to interface it with an industrial controller.

8.2.2 Test bench requirements for the control platform (software)

The test bench on for the control platform side has the following requirements:

- To have an integration of high level programming languages (such as C++)
- Supports C++ code generated from MATLAB Coder
- Provides high speed real time cyclic execution of the algorithm
- Integrates with other industrial components such as industrial robots and the remaining automation set up
- Should require minimum additional hardware in addition to the hardware which is already in the assembly line.

- Provides industrial interfaces such as ProfiNET, ProfiBUS etc. for integration of the I/O as well as for direct communication with other actuators and sensors.
- The Controller is designed in Simulink and the targeting platform should be supported so that Simulink can connect to it as an external target
- Provides cycle times of up to 1ms to process signals from an encoder to measure the tightening angle.

To do this, three options have been analysed. There are many more platforms, however, the most common and suitable systems have been analysed.

8.2.1 Using an I/O Card within MATLAB and Real-time Linux

The first option which has been analysed is the use of an I/O card (for example the Sensoray Model 626 [74] in combination with the real time target of MATLAB within a real time Linux environment. The integration is rather simple as MATLAB supports this I/O Card and also drivers for Linux are available. The I/O card allows to integrate a tightening tool and the system also provides the capability of running the control algorithm on a fast cyclic speed. Industrial interfaces (such as ProfiNET, ProfiBUS etc.) can also be integrated so that other sensors and actuators can be accessed.

Currently Linux applications in Industry play a role for individual applications (for example mobile robots) but considering that modern assembly lines are based on a classical PLC structure, where the control strategy needs to be integrated, the Linux and the I/O strategy may not be suitable. Furthermore, the I/O card needs to provide an industrial standard to ensure that there is no failure for several years of permanent operation. Therefore, the control algorithm should target an industrial standard control environment ideally where the hardware is already available in factories and the integration process is minimized.

8.2.2 Using a Phoenix Contact PLC and an Industrial PC for executing the control strategy

A Phoenix Contact [59] setup as also been analysed for the control application. The setup which has been provided within the COSMOS Project cooperation contains a PLC and an industrial PC based on Windows XP running PCWorx, which is an automation software from Phoenix Contact. Continuously, other industrial components within the environment such as a robot can be integrated and also controlled. The MATLAB / Simulink model can be converted into real time C++ Code by using MATLAB Coder and run on the industrial PC where it is executed on highest priority in Windows XP.

This setup is able to run the control algorithm as well as provides support for all the necessary BUS Systems (ProfiNET, ProfiBUS etc.). The PLC can be connected via Ethernet to the industrial PC and a communication needs to be set up to read and write the inputs and outputs of the PLC. The cycle time of the PLC can be set up to 1ms. The inputs and outputs of the PLC can be configured in any desired way since they are based on terminal blocks and can be clocked next to each other in any possible way.

In addition, Windows is not a real time system and therefore, hard real time conditions may not be guaranteed but are required for the control application. Therefore, this system may not perform on a suitable level as it provides risks with using C++ Code generated by MATLAB Coder. Another option would be using MATLAB Coder to generate PLC Code (Structured Text) and run it on the Phoenix Contact PLC. Further investigation turned outlined that the support of the Simulink block library is rather limited. Also, it would be good for the development of the controller to connect to the Simulink target while the controller is running. This may not be guaranteed with this system.

8.2.3 Using a Beckhoff TwinCAT 3 System

Finally, a Beckhoff System [58] has been analysed for the setup. The Beckhoff TwinCAT 3 runs on cycle speeds up to 10µs and therefore provides enough speed for the execution of the control algorithm. Similarly to the Phoenix Contact system, the I/O can be expanded by using I/O terminals which can be clicked next to each other.

Continuously, other components such as a Fanuc robot can be accessed using standard interfaces (ProfiNET, ProfiBUS etc.). The Software runs on an industrial PC which is already available in the assembly line and therefore this requires only a software extension plus adding the I/O.

8.3 Simulink Model for the four stages bolt tightening controller (chapter 2)

The four stages tightening controller is based on a Simulink model which has been converted with MATLAB Coder to real time C++ Code, which has been imported into Beckhoff TwinCAT 3. TwinCAT then executes the controller in real time cyclically with a cycle time of $\mathrm{T} = 500 * 10^{-6}$s.

The Simulink Model is a nested model; its structure is shown on the next pages. As it is shown in Figure 68 the controller has several sub blocks. The actual four stages fuzzy controller is in the '4 Stages fuzzy controller' block. The other blocks contain functions to store the error and stop the controller, convert the current readings into actual torque values and also there is a block with a main flip flop keep the controller activated or stopped. Furthermore, there is the essential emergency stop function. This will stop the whole controller module and requires a restart after the emergency stop has been activated. Furthermore, the switching between the 4 stages is also done with one control block. This is done as soon as a controller has reached its target.

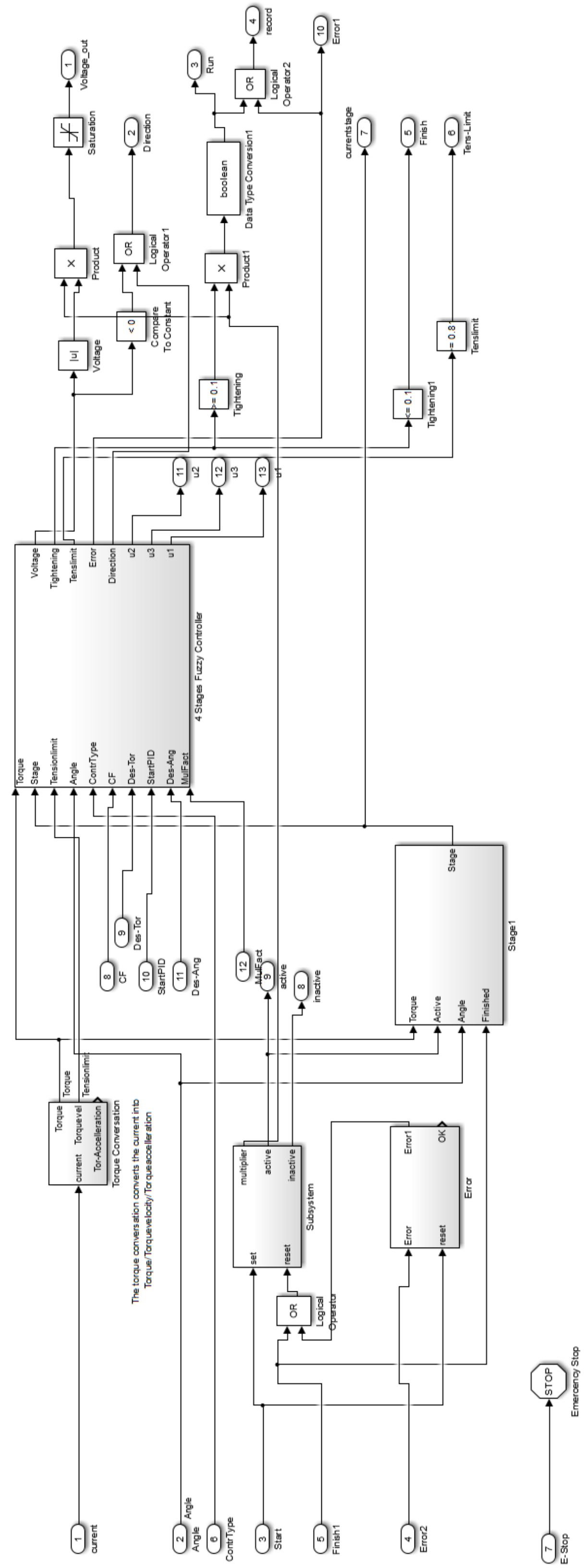


Figure 68: Simulink Model overview for the four stages fuzzy controller

The next block is the four stages fuzzy controller block. It contains the actual fuzzy controller as well as the PID controller which has been used for comparison of the fuzzy controller performance and the PID controller performance. The block is shown below:

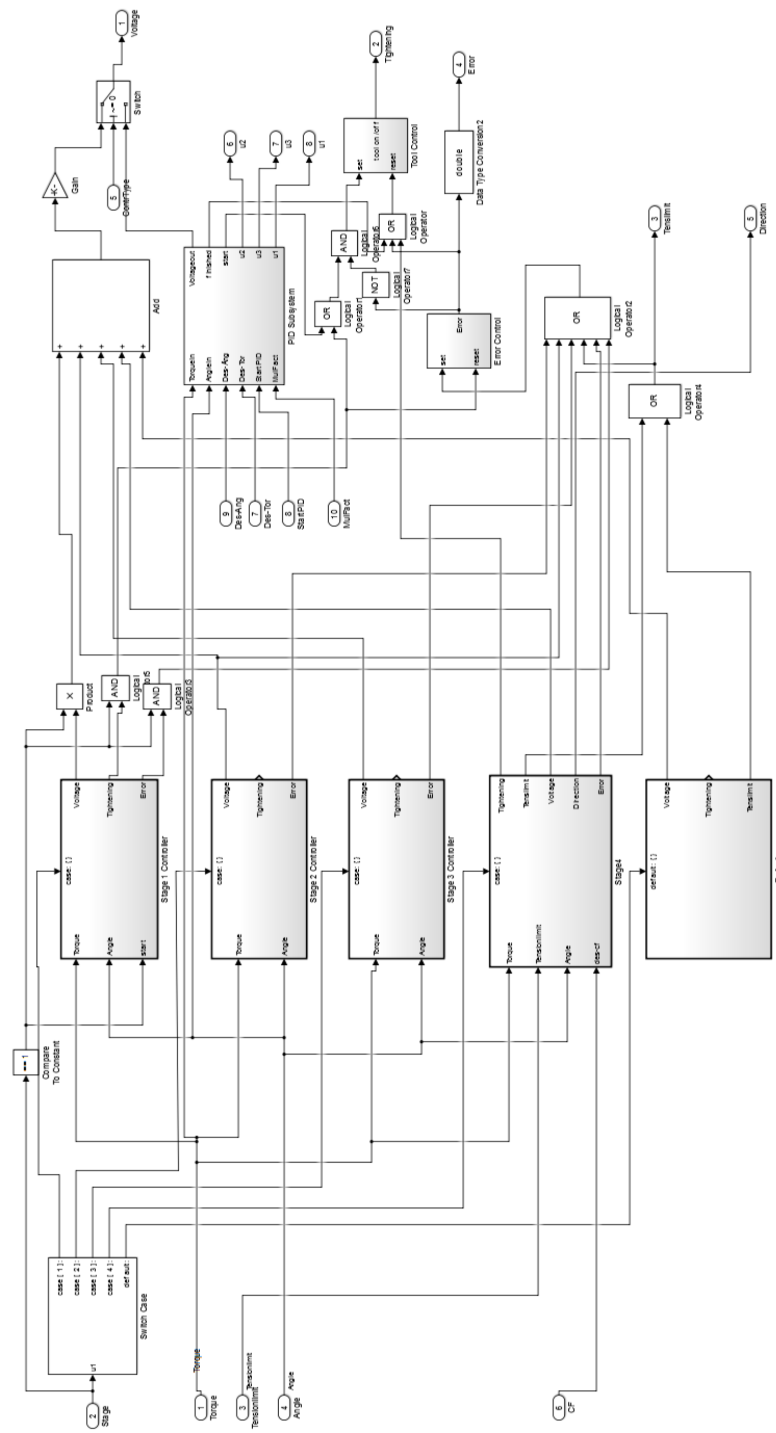


Figure 69: 4 stages controller block

Each Fuzzy controller contains the individual requirements for a stage, as described in Chapter 2. The stages can only executed in a sequential order, it is not possible to execute the stages in reverse order.

The internal set up showing each stage is provided in the next figures.

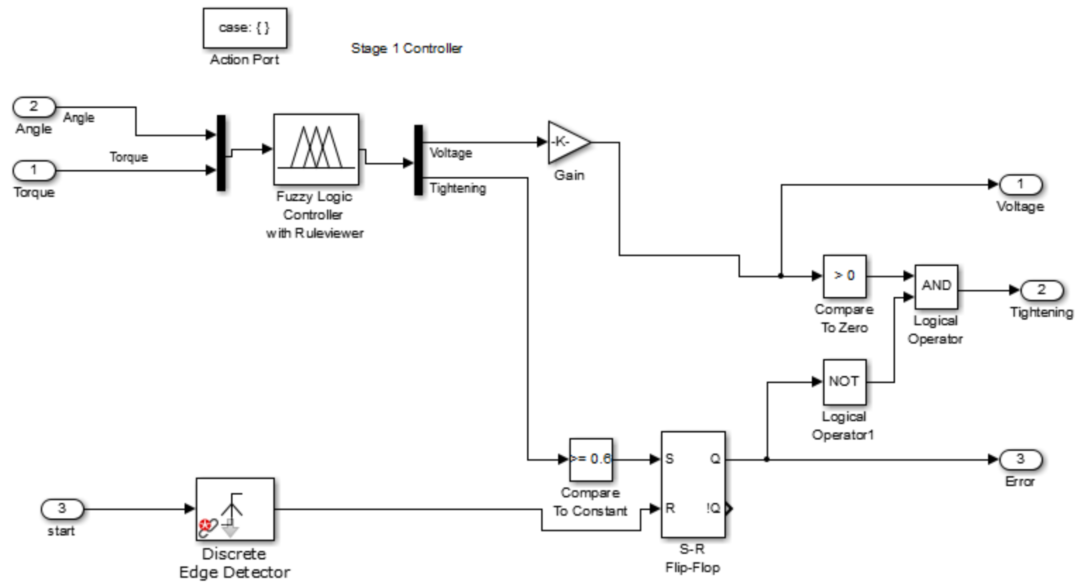


Figure 70: Stage 1 fuzzy controller

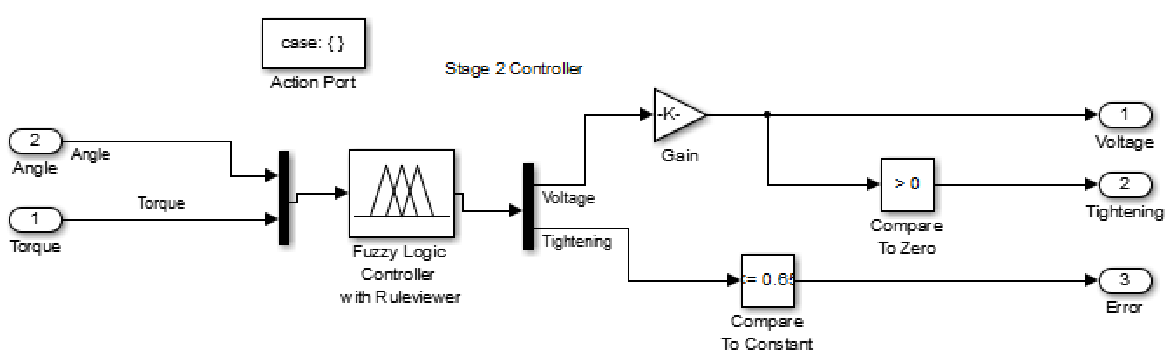


Figure 71: Stage 2 fuzzy controller

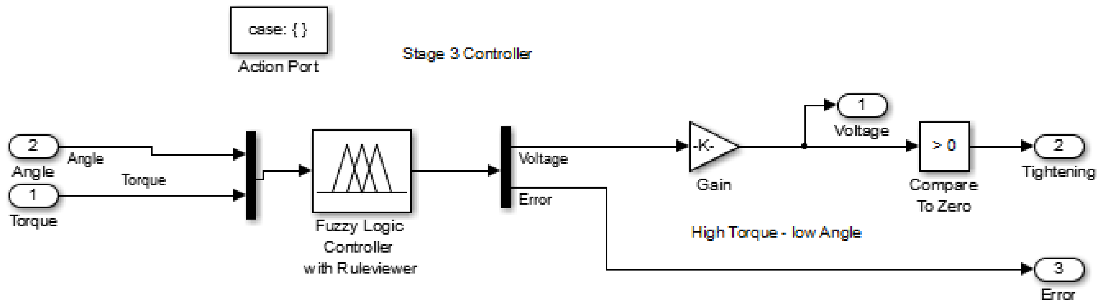


Figure 72: Stage 3 fuzzy controller with error detection

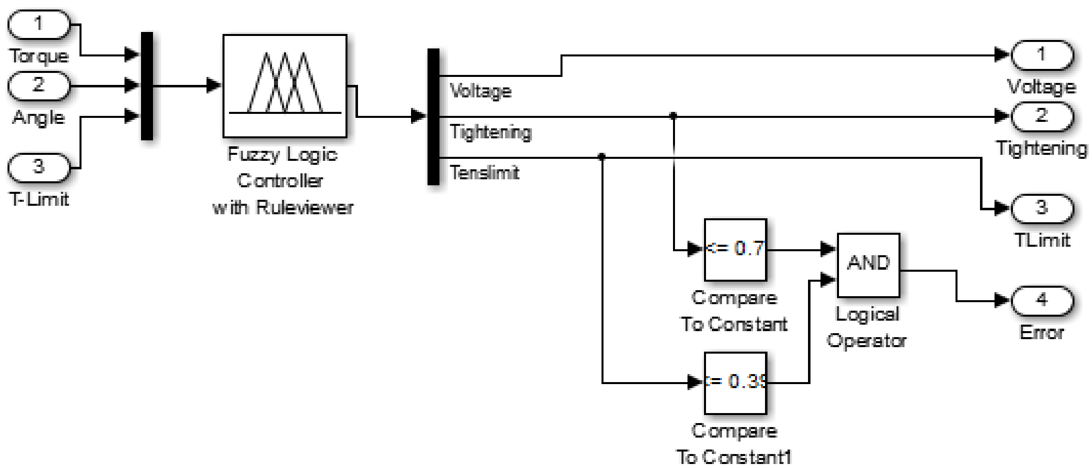


Figure 73: Stage 4 fuzzy control set up

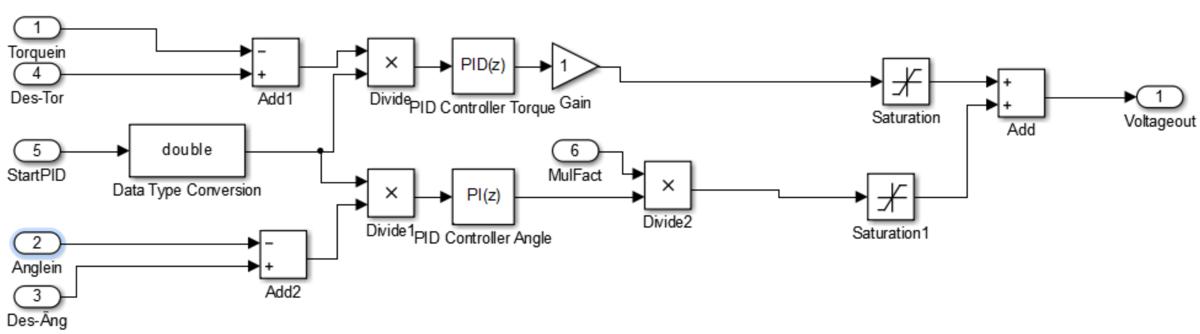


Figure 74: Discrete PID Controller for comparison of the fuzzy controller

Figure 74 shows the used model for the PID Controller. The control design for the PID Parameter have been based on a classical control. It is based on the torque/angle tightening technique [16].

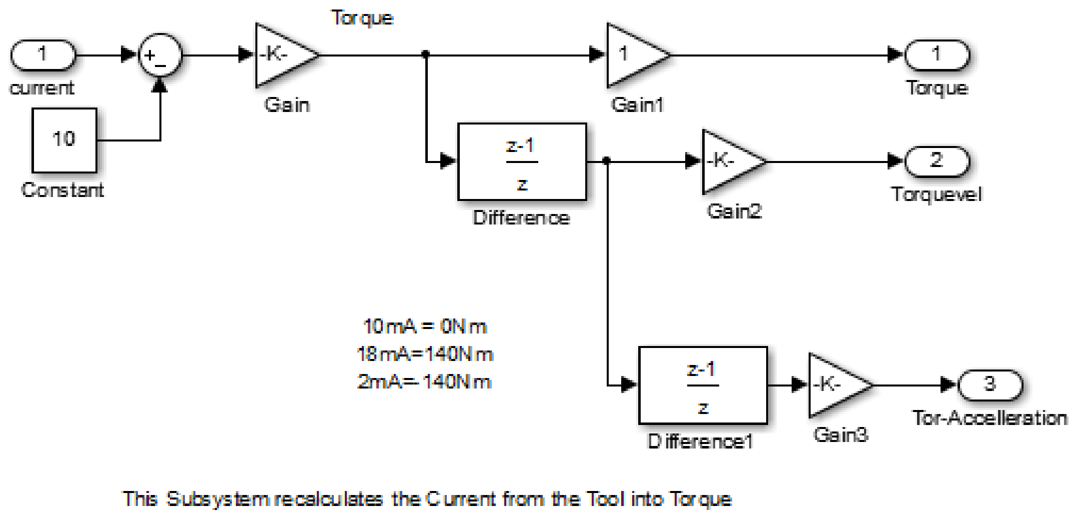


Figure 75: Torque conversion block

The torque block converts the raw torque signal which is a 16 bit analogue signal into current and then into torque. Furthermore, the velocity and the acceleration has also been estimated. This is used for the error detection functionality. In particular, it is essential to detect the tension limit of the bolt. This is reached when the velocity becomes constant. The tightening process has to be stopped immediately then to prevent any physical damage.

8.4 Self PI tuned Algorithm

The algorithm introduced in Chapter 3 is using a Simulink model as well as a Matlab program. The model is nested again and based on the previous 4 stages fuzzy controller. However, the fuzzy controller has been removed and the PID controllers Gains are changeable.

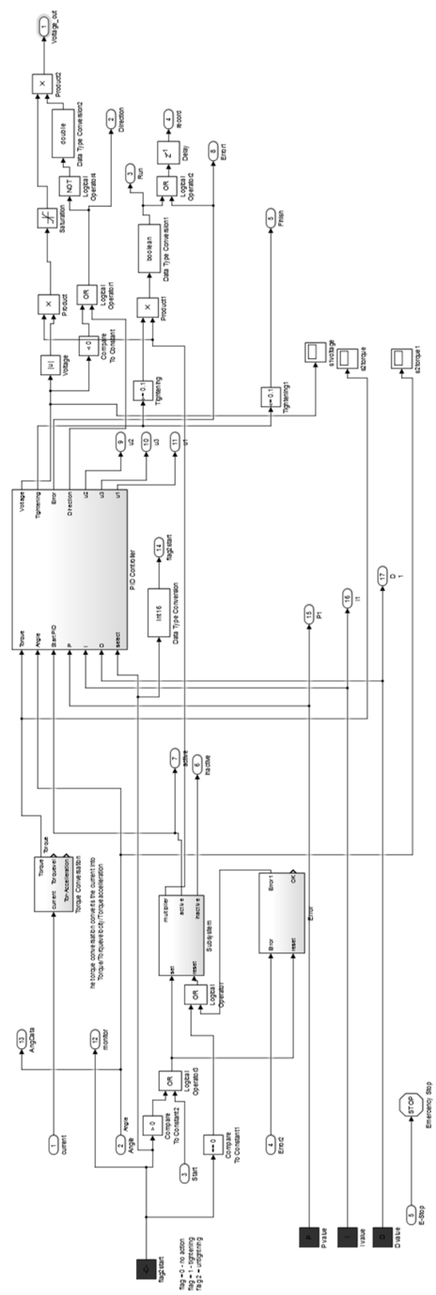


Figure 76: Model based PI Tuning block

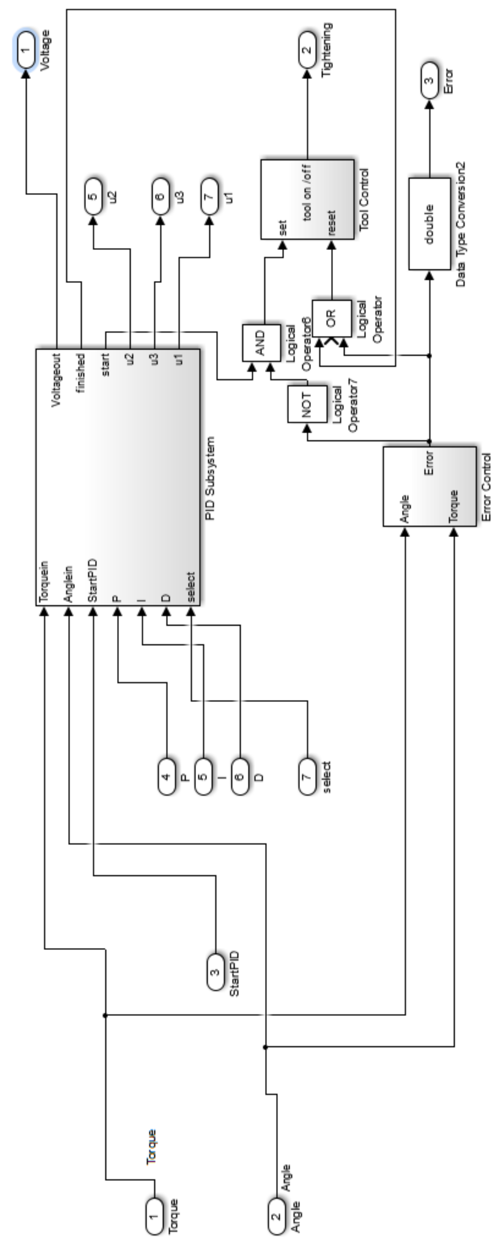


Figure 77: PI controller block

Figure 77 shows the actual controller block which is divided into the error recognition block, which contains comparison blocks to detect faults based on torque and angle values as described in Table 7. Furthermore, the internal structure is shown in Figure 78.

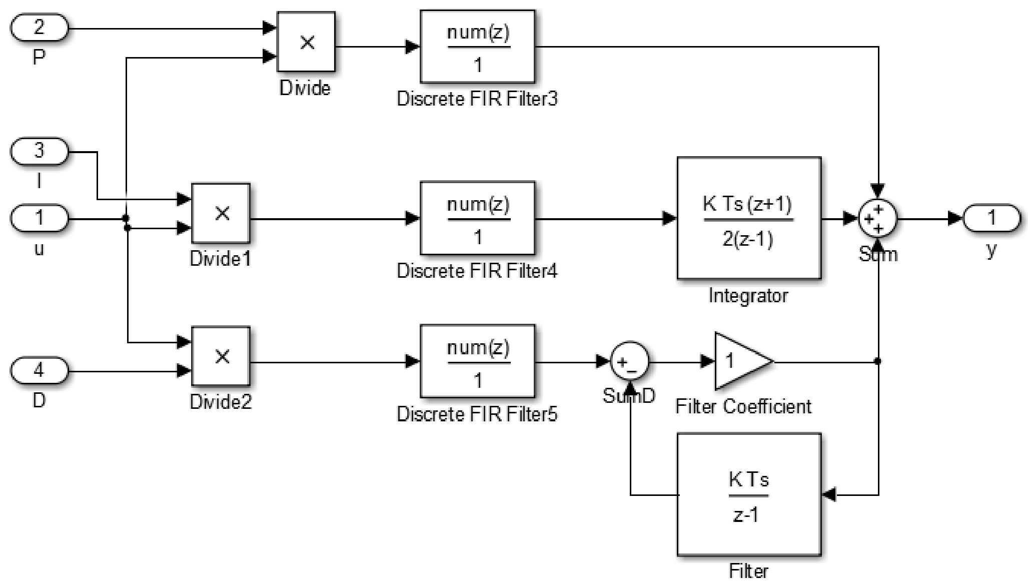


Figure 78: Internal discrete PID controller

The gains can be externally tuned and have been set as tuneable variables. The step response data is recorded with a scope which is also set that the data can be collected from the MATLAB program after completion of the algorithm.

The implementation purely based on Simulink leads to the problem that MATLAB coder cannot convert all functions required for the Genetic Algorithm into real time C++ code. Therefore, the program has been changed in a way, that the PI Controller is executed in real time (cycle time $C_T = 500 * 10^{-6}$s) and the result of the control run is send to a MATLAB program which implements the genetic algorithm. The procedure is shown in Figure 79.

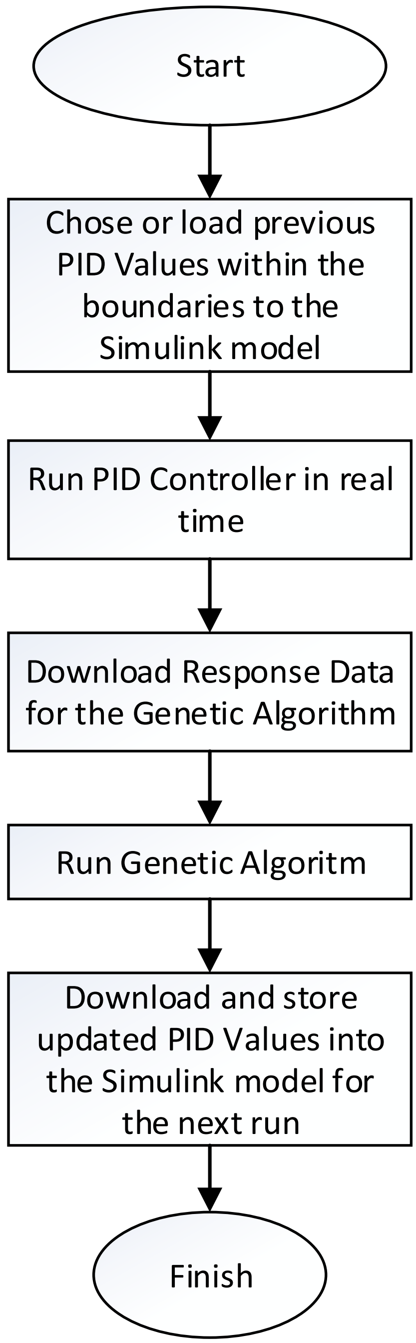


Figure 79: Execution of the GA combined with the physical system

When the process is started, a MATLAB program is executed which either loads the model estimated PI values or loads previous PI values based on the model estimated values which already have been fine-tuned with the Genetic Algorithm. The Simulink model needs to be started on the Beckhoff controller and connected as an external target to Simulink. The PI values are then uploaded into the Simulink model and the controller is started. Once the tightening is completed the step response data is copied from the real time model and processed within the GA to fine-tune the PI values further within the boundary setting. It should be considered that the GA is always active, even though the step response may not be further improved. In this situation, if the performance criteria of the GA has been matched, the same PI values will be used until the step response changes again. This allows to learn the attributes of a set of bolts and apply them until the set us installed and a new set is introduced to the assembly line. The response data of the new bolt will be different than from the previous set and the GA will learn again the non-linearity and uncertainties of the new set of bolts.

For the experiments, the bolt bench has been used to record the data and to allow to run the experiments autonomously.

9. Appendix B: torque reaction system

Appendix B provides technical details, such as technical drawings about the clamp system and the bolt bench for multiple tests of the tightening algorithms.

The main aim of the torque reaction system is to prevent any backward torque going into the robot arm. The robot is using joints which are not set up to accept more than the original torque applied from the robot arm itself. The reaction system provides a closed system where the torque is locked up into during the assembly process. This minimized the additional torque level on the robot arm and furthermore, any additional forces are also only applied in the closed torque reaction system.

To prepare the experimental set up, a bolt bench has also been designed. A real wind turbine hub was not available for testing; furthermore, the existing FANUC M6iB robot is also too small for the final tests.

9.1 Appendix B Structure

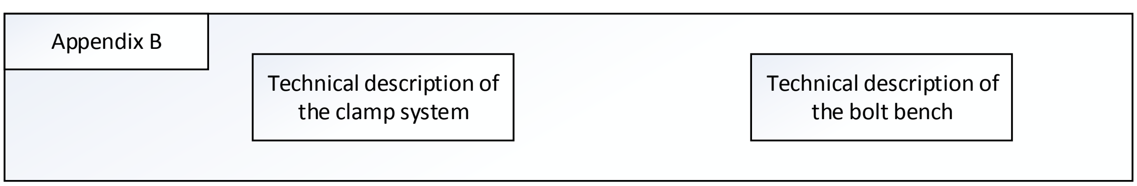


Figure 80: Appendix B Structure

9.2 Clamp Drawings

The torque reaction drawings contain the bolt bench (bolt plate), the horizontal plate, the middle support and the vertical plates.

9.2.1 Drawing: Bolt Plate

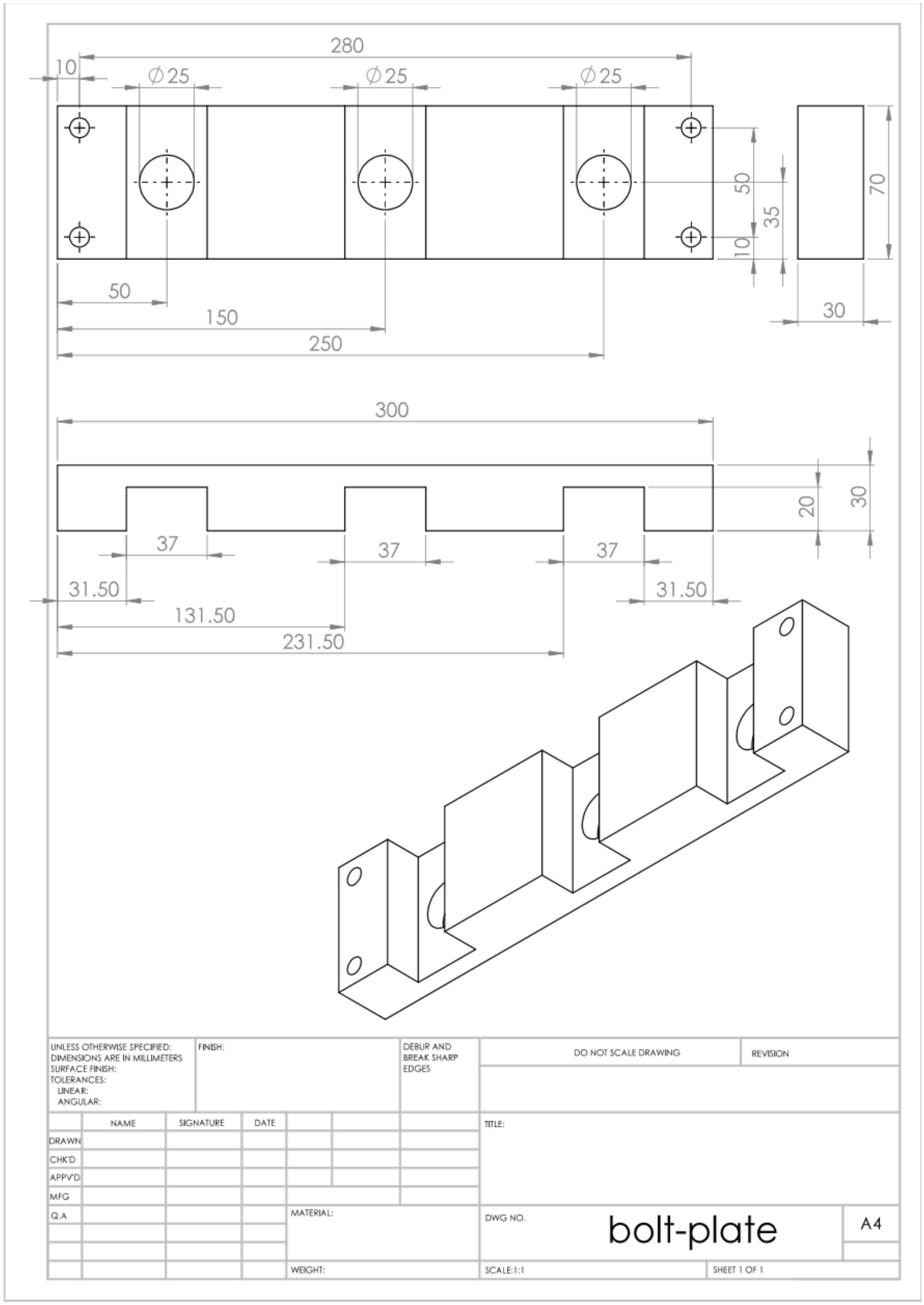


Figure 81: Bolt Plate

9.2.2 Drawing: Horizontal Plate

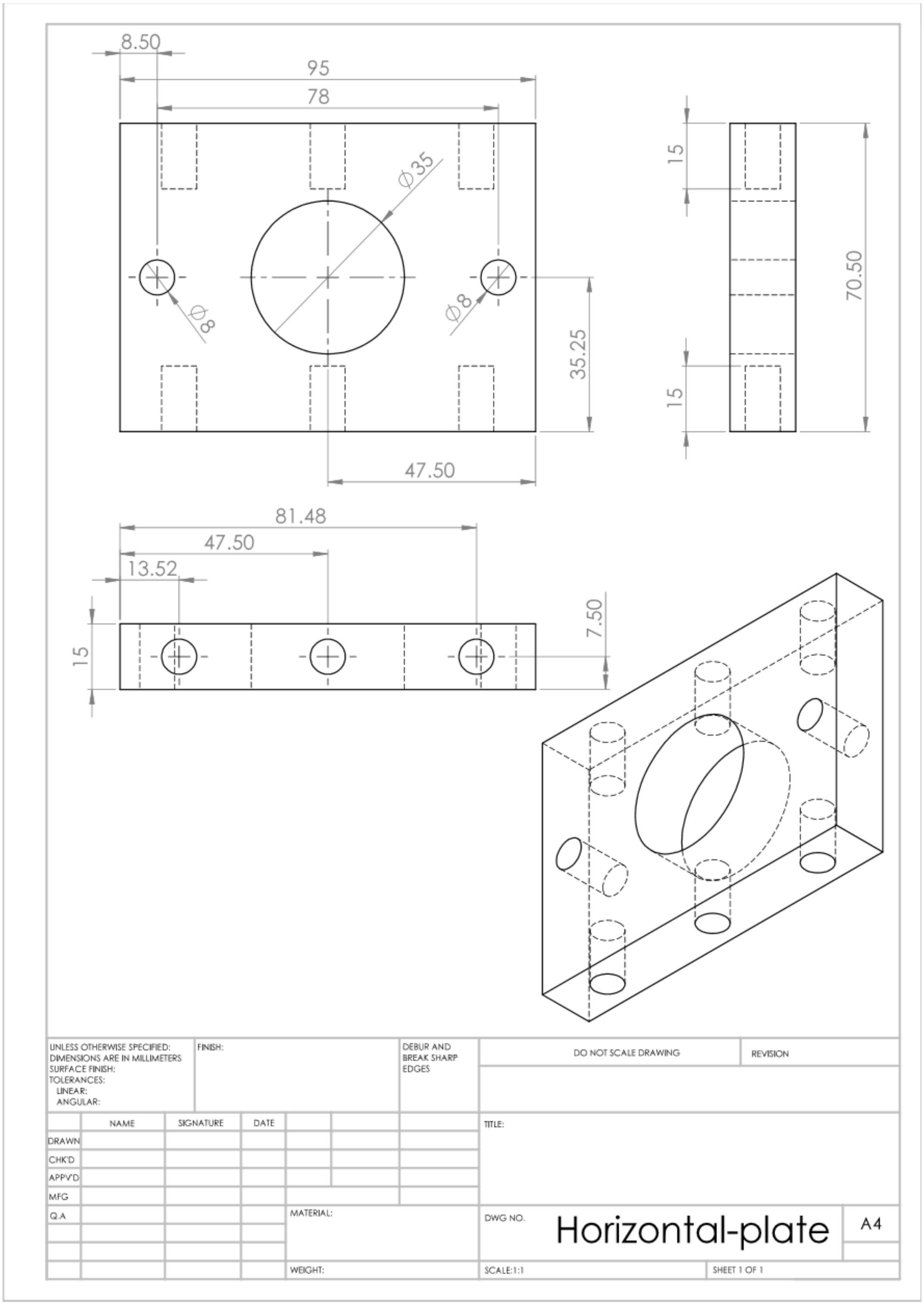


Figure 82: Horizontal Plate

9.2.3 Drawing: Middle Support

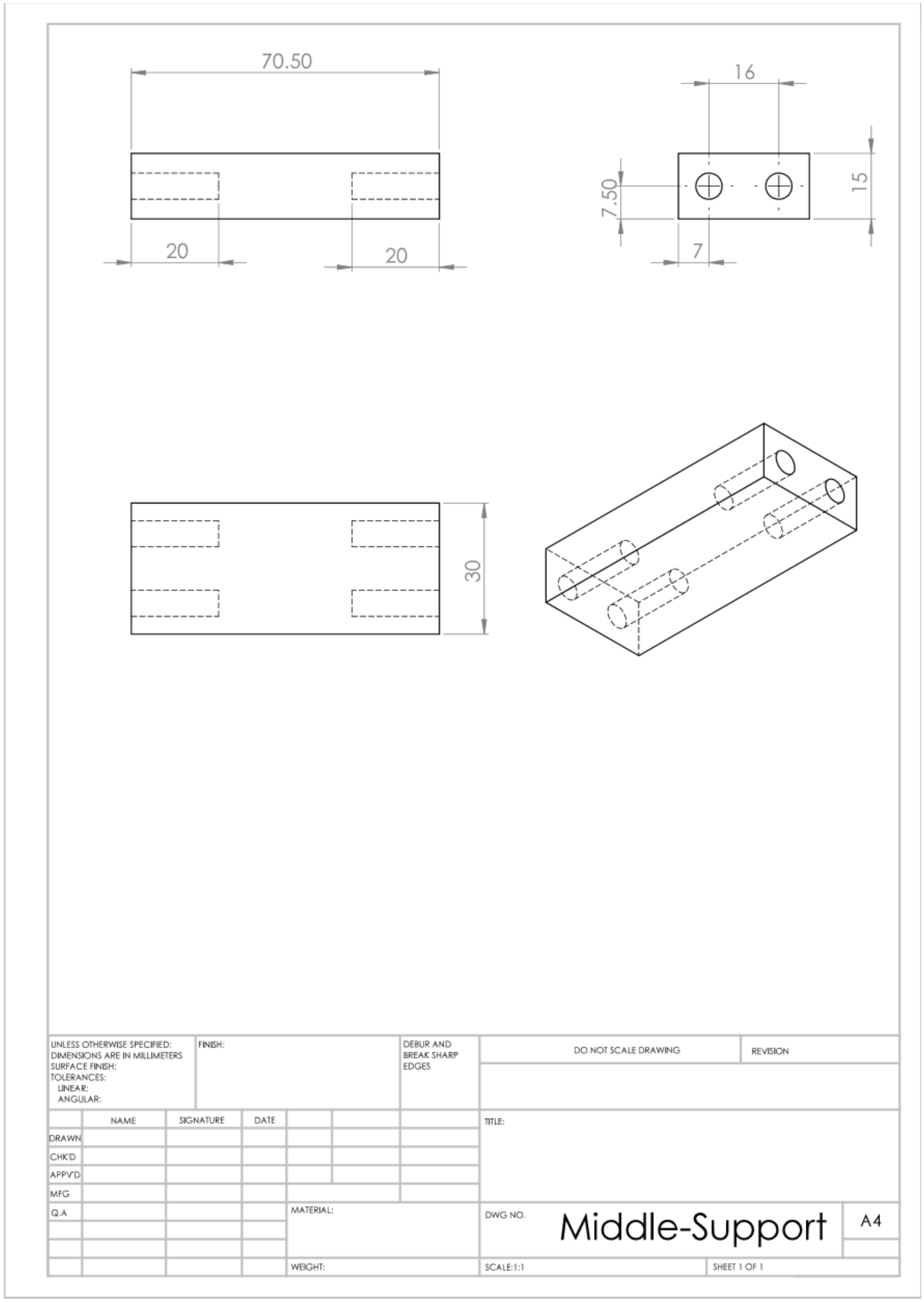


Figure 83: Middle Support

9.2.4 Drawing: Vertical Plates

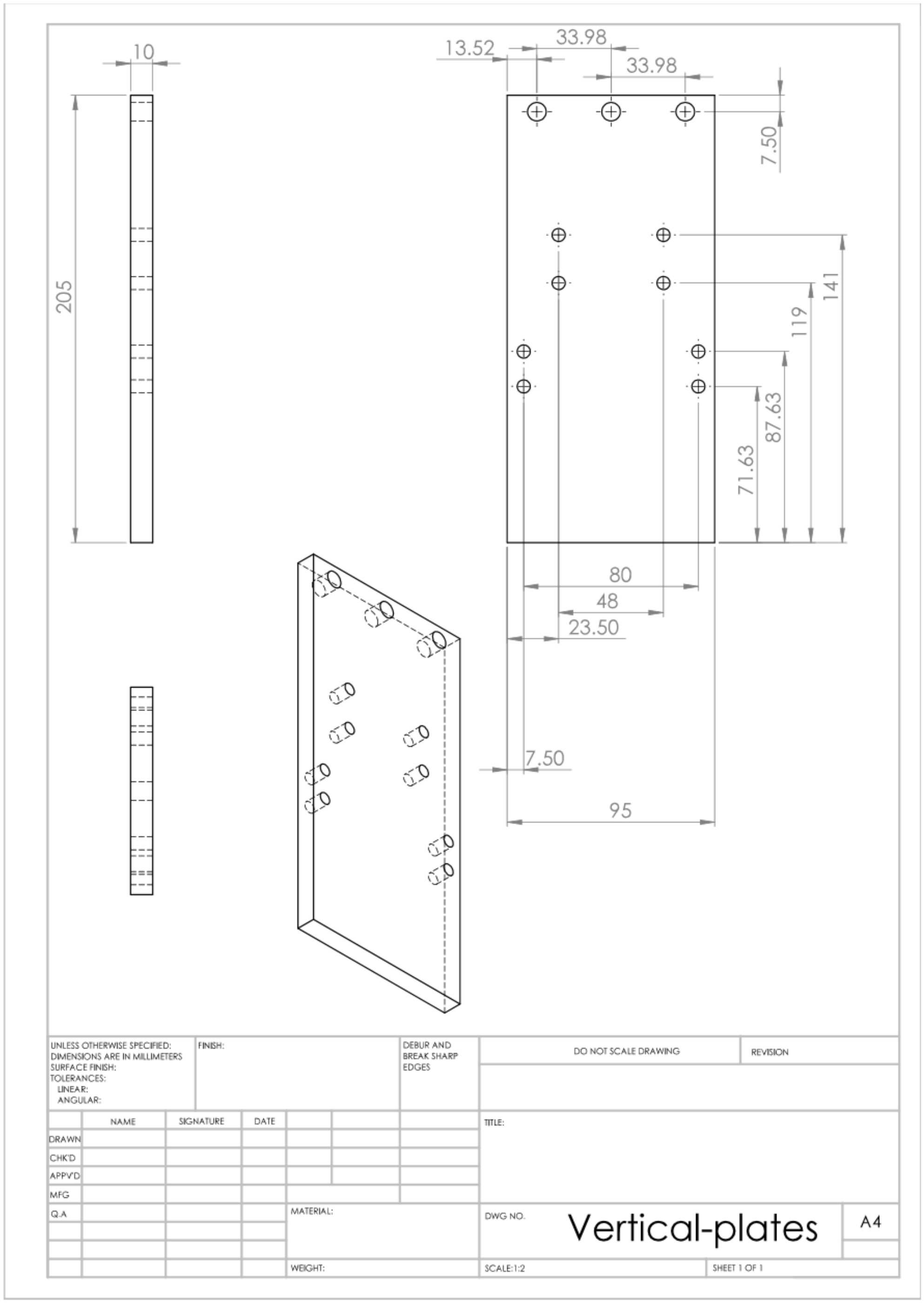


Figure 84: Vertical Plates

9.3 Bolt tightening bench

The bolt test bench is shown in the picture below. It has been designed at a workshop directly, therefore, no technical drawings are provided. The main aim is that several experiments can be carried out automatically and the clamping force can be measured at the same time. To do this, a clamping force sensor has been installed on the bolt system. Furthermore, the bench allows also to install several different bolt sizes. The focus on this research has been on a M24 bolt, which is a common size for a mid-ranged turbine from Gamesa.

The next pictures show the bench without the clamping force sensor:

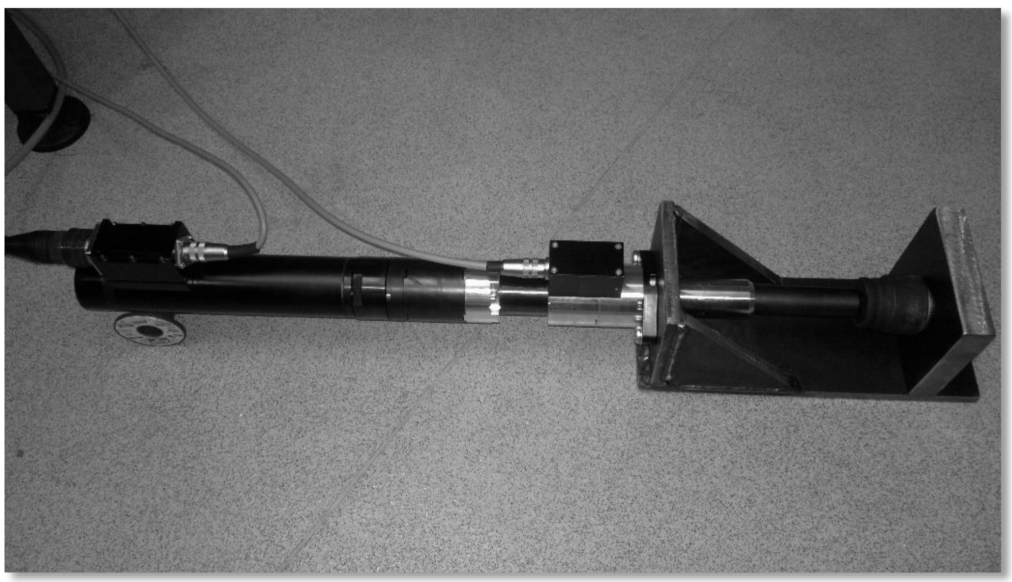


Figure 85: Tightening tool in the bench without the clamping force sensor

Figure 86: Picture of the tool in the bench at the nut

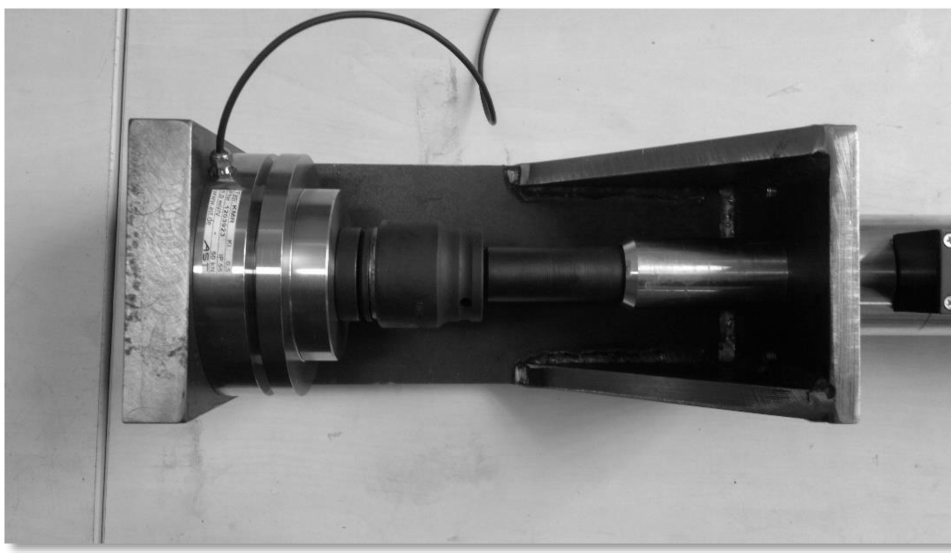


Figure 87: Tool with the clamping force sensor

The clamping force sensor [75] is a ring sensor which measures clamping forces up to 50KN. It can be directly connected to the Beckhoff I/O System.

This setup provides stability to ensure that even on a faulty algorithm set up where the tightening tool runs on maximum speed and maximum torque where safety features may fail the set up stays stable. Therefore, before the algorithms were applied within the robotic set up, they were extensively tested on this bench.

10. References

[1] S. Breton, G. Moe "Status, plans and technologies for offshore wind turbines in Europe and North America," *Renewable Energy 34 ,* p. 646–654, 2009.

[2] Fang Yao, R. C. Bansal et al., Wind Energy Resources: Theory, Design and Applications, World Scientific Publishing.

[3] C. S. Ezio, "Exploration of wind energy as an energy source to meet the worlds electricity demand," *Wind Eng. ,* vol. 74, no. 76, pp. 375-387, 1998.

[4] "Research and Innovation Energy," European Commision, [Online]. Available: http://ec.europa.eu/research/energy/eu/index_en.cfm?pg=research-wind. [Accessed 11 2013].

[5] Tony Burton, Wind Energy Handbook, Wiley, 2001.

[6] M. J. K. Brian Snyder, "Ecological and economic cost-benefit analysis of offshore wind energy," *Renewable Energy,* vol. 34, no. 34, pp. 1567-1578, 2009.

[7] R. Saidur, "A review on global wind energy policy," *Renewable and Substainable Energy Reviews,* vol. 14, no. 14, pp. 1744-1762, 2010.

[8] Z Chen M. Guerrero, F. Blaaberg, "A Review of State of the Art of Power Electronics for Wind Turbines," *IEEE Transactions on Industrial Electronics,* vol. 24, no. 8, pp. 1859-1875, 2009.

[9] Suzlon Energy, Interviewee, *Manufacturing of wind turbines.* [Interview]. 15 08 2010.

[10] C. Deters Helge Wurdemann and Kaspar Althoefer, "Reconfigurable Assembly Approach for Wind Turbines using Multiple Intelligent Agents," in *ASME/IEEE International Conference on Reconfigurable Mechanisms and Robots*, Tianjin, China.

[11] H.K. Lam et al, "Control Design for Interval Type-2 Fuzzy Systems Under Imperfect Premise Matching," *IEEE Transactions on Industrial Electronics,* vol. 61, no. 2, pp. 956-968, 2013.

[12] P. Maegaard, "Nordic Folkecenter," [Online]. Available: http://www.folkecenter.net/mediafiles/folkecenter/awards/XEMC-Wind-billeder-%2821%29.jpg. [Accessed 11 2013].

[13] "Wind Turbine Accident Data," Caithness Wind Farm Information Forum, [Online]. Available: http://www.caithnesswindfarms.co.uk/AccidentStatistics.htm. [Accessed 05 02 2014].

[14] "Telegraph," [Online]. Available: http://i.telegraph.co.uk/multimedia/archive/01218/UFO_1218501f.jpg. [Accessed 11 2013].

[15] "ISO Darstellungen Schrauben," [Online]. Available: http://diglib.ethz.ch/system/temporary/get_for.ind6.de.htm.

[16] SKF, Bolt tightening Handbook, 2001.

[17] H Wittel, "Schraubverbindungen," in *Maschinenelemente*, Vieweg+Teubner Verlag, 2011.

[18] "Schrauben Lexikon," [Online]. Available: http://www.schrauben-lexikon.de/index.asp. [Accessed 11 2013].

[19] Takehirp Onodera, "Automotive Bolts Tightening Analysis using Contact Stress Simulation: Developing an Optimal CAE Design Approach Model," *Journal of Business & Economics Research,* vol. 10, no. 7, pp. 435-442, 2012.

[20] M. Sharpe, "Robotic Fabrication of Wind Turbine Power Generators," Fanuc Robotics, 2009.

[21] C. Deters, "Model free Fuzzy Tightening Control for Bolt/Nut Joint Connections of Wind Turbine Hubs," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013.

[22] Yokoyama et al., "Investigation into the self-loosening behaviour of bolted joint subjected to rational loading," *Engineering Failure Analysis,* vol. 23, pp. 35-43, 2012.

[23] Hicham Chaoui, "Adaptive Fuzzy Logic Control of Permanent Magnet Synchronous Machines With Nonlinear Friction," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 2, pp. 1123-1133, 2011.

[24] Johanes H. J. Potgieter, "Torque and Voltage Quality in Design Optimization of Low-Cost Non-Overlap Single Layer Winding Permanent Magnet Wind Generator," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 5, pp. 2147-2156, 2012.

[25] Y. Maeda, "Initial Friction Compensation Using Rheology Based Rolling Friction Model in Fast and Precise Positioning," *IEEE Transactions on Industrial Electronics,* vol. 60, no. 9, pp. 3865-3876, 2012.

[26] A. Talebi, "A Neural Network-Based Multiplicative Actuator Fault Detection and Isolation of Nonlinear Systems," *IEEE Transactions on Control Systems Technology,* vol. 21, no. 3, 2013.

[27] A. Dirksz, ""Power-Based Set Point Control: Experimental Results on a Planar Manipulator," *IEEE Transactions on Control Systems Technology,* vol. 20, no. 5, 2012.

[28] K. H. Ang, G. Chong, "PID control system analysis, design and technology," *IEEE Transactions on Control Systems Technology,* vol. 13, no. 4, 2005.

[29] N. Dhayagude, Z. Gao, F. Fouad, "Fuzzy Logic Cointrol of Automated Screw Fastening," *Robotics & Computer Integrated Manufacturing,* vol. 12, no. 3, pp. 235-242, 1996.

[30] S Izumi, T. Yokoyama, M. Kimura, "Loosening-resistance evaluation of double-nut tightening method and spring washer by three dimensional finite emement analysis," *Engineering Failure Analxsis,* vol. 16, pp. 1510-1519, 2009.

[31] Sung-Pil Jung, "Design Optimization of Spring of a Locking Nut using Design of Experiments," *International Journal of Precision Engineering and Manufacturing,* vol. 10, no. 4, pp. 77-83, 2009.

[32] J. Jantzen, "Tuning Of Fuzzy PID Controllers," Technical University of Denmark, 1998.

[33] T. Matsuno, "Fault detection Algorithm for External Thred Fastening by Robotic Manipulatior Using Linear Support Vector Machine Classifier," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[34] K. Althoefer, B. Lara, Y.H. Zweiri, L.D. Seneviratne, "Automated failure classification for assembly with self-tapping threaded fastenings using artificial neural networks," *Mechanical Engineering Science,* vol. 222, 2008.

[35] T. Fujinaka, H. Nakano, S. Omatu, "Bolt tightenign Control using neural networks," in *IEEE Conference on Systems, Man and Cybernetics*, Tucson, AZ, 2001.

[36] J. L. Verdegay, E. Vergara-Moreno, “Fuzzy Sets-based Control Rules for Termination Algorithms,” *Computer Science Journal of Moldova,* vol. 28, no. 1, pp. 9-26, 2002.

[37] Y. Fan, C. Guo, W. Meng., “Study of Distributed Multiaxial Intelligent Tightening Machine BAsed on Fieldbus Control System,” in *International Conference on Intelligent Control and Information Processing*, Dalian, China, 2010.

[38] P. Yuan, “An Adaptive Feedback Scheduling Algorithm for Robot Assemblöy and Real Time Control Systems,” in *International Conference on Intelligent Robots and Systems*, Beijing, 2006.

[39] “Wind Turbine Design,” Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Wind_turbine_design.

[40] “Würth Website,” Würth, 2014. [Online]. Available: http://www.wuerth.de/web/de/orsyfleet/produkte_1/drehmomentschluessel/drehmomentschluessel_1.php. [Accessed 01 04 2014].

[41] “Bahco,” Bahco, [Online]. Available: http://www.bahco.com/english/. [Accessed 02 04.2014].

[42] “DSM Messtechnik,” DSM Messtechnik, 02 08 2011. [Online]. Available: http://www.dsm-messtechnik.de/english/.

[43] R. Vargas, “Predictive Torque Control of an Induction Machine Fed by a Matrix Converter With Reactive Input Power Control,” *IEEE Transactons on Power Electronics,* vol. 25, no. 6, pp. 1426-1438, 2010.

[44] M. Korkmaz, “Design and Performance Comparison of Variable Parameter Nonlinear PID Controller and Genetic Algorithm Based PID Controller,” in *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2012.

[45] A. Linkens, “Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications,” *IEEE Proceedings in Control Theory Applications,* vol. 143, no. 4, pp. 367-386, 1995.

[46] T. Zhang, “A Methodology for Block-Orientated Industrial Nonlinear System by Nonlinear Separation Control with Neural Learning,” in *SICE*, 2002.

[47] C. Deters, H. K. Lam, E. Secco, H. A, Wurdemann LD Seneviratne, K. Althoefer, "Accurate Bolt Tightening using Model-Free Fuzzy Control for Wind Turbine Hub Bearing Assembly," *IEEE Transactions on Control Systems Technology,* 2014.

[48] S. Hajri, "A Controlled Genetic Algorithm by Fuzzy Logic and Belief Functions for Job-Shop Scheduling," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS,* vol. 30, no. 5, pp. 812-818, 2000.

[49] C. Deters, H. K. Lam, E. Secco, H.A. Wurdemann, K. Althoefer, "Model Based Self-Tuning PI Control of Bolt-Nut Tightening for Wind Turbine Bearing Assembly," *IEEE Transactions on Automation Science and Engineering,* 2014.

[50] V. Rutvij C. Joshi, "Congestion Control in Communication Networks Using Discrete Sliding Mode Control," Hefei, China, 2012.

[51] Frank H. F. Leung, "Optimal and Stable Fuzzy Controllers for Nonlinear Systems Based on an Improved Genetic Algorithm," *IEEE Transactions on Industrial Electronics,* vol. 51, no. 1, pp. 172-182, 2004.

[52] J. Villagra, V. Milanes, J. Perez, C. Gonzalez, "Model free control techniques for Stop & Go Systems," *IEEE Annual Conference on Intelligent Transportation Systems,* pp. 1899-1904, 2010.

[53] R. Marden, S.D. Ruben, L.Y. Pao, "A Model-Free Approach to Wind Farm Control Using Game Theoretic Methods," *IEEE Transactions on Control Systems Technology,* 2013.

[54] B. Xuhui, "A Statistical Analysis of Model Free Adaptive Control with Measurement Disturbance," in *Chinese Control Conference*, Beijing, 2010.

[55] H. Hanao, "Wire Rope Fault detection in a Hoisting Winch System by Motor Torque and Current Signatuire Analysis," *IEEE Transactions on Industrial Electronics,* vol. 58, no. 5, pp. 1727-1736, 2011.

[56] B. Akin, S. Choi, U. Orguner, H.A. Toliyat, "A Simple Real-Time Fault Signature Monitoring Tool for Motor-Drive-Embedded Fault Diagnosis Systems," *IEEE Transactions on Industrial Electronics,* vol. 58, no. 5, pp. 1990-2001, 2011.

[57] T. Thumati, G.R. Halligan, "A Novel Fault Diagnostics and Prediction Scheme Using a Nonlinear Observer With Artificial Immune System as Online Approximatior," *IEEE Transactions on Control Systems Technology,* vol. 21, no. 3, 2013.

[58] Beckhoff, "Beckhoff," Beckhoff, 2013. [Online]. Available: www.beckhoff.de. [Accessed 2011].

[59] Phoenix Contact, "Phoenix Contact PC Worx," Phoenix Contact, [Online]. Available: www.phoenixcontact.com. [Accessed 2013].

[60] F. Mrad, Z. Gao, N. Dhayagude, "Fuzzy logic control of automated screw fastening," *Robotics & Computer-Integrated Manufacturing,* pp. 235-242, 1996.

[61] N.G. Pai, D.P. Hess, "Three-dimensional finite element analysis of tightening and loosing mechanism of threaded fastener," *Engineering Falure Analysis,* vol. 12, no. 4, pp. 604-615, 2005.

[62] A. Mintsa, R. Venuopal, J.-P- Kenne, B. Belleau, "Feedback Linearization-Based Position Control of an Electrohydraulic Servo System With Supply Pressure Uncertanty," *IEEE Transactions on Control Systems Technology,* vol. 20, no. 4, 2012.

[63] L.D. Seneviratne, et al., "Theoretical modelling of the self-tapping screw fastening process," in *Proceedings of the international conference in intelligent Robots and Systems (IROS)*, 2003.

[64] J. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, Michigan: University of Michigan Press, 1975.

[65] H.-W. Philippsen, Einstieg in die Regelungstechnik: Vorgehensmodell für den praktischen Reglerentwurf, Bremen: Fachbuchverlag Leipzig, 2006.

[66] Steven R. Haynes, "Designs for explaining intelligent agents," *International Journal of Human-Computer Studies,* vol. 67, pp. 90-110, 2009.

[67] Gael Clair, "Self-Regulation in Self-Organizing Multi-Agent Systems for Adaptive and Intelligent Manufacturing Control," in *Secound IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Venezia, 2008.

[68] Murat Sensoy, "Flexible Task Resourching for Intelligent Agents," in *International Conference on Autonomous Agents and Mulitagent Sysems*, Toronto, Canada, 2010.

[69] Karim Ramirez, "Monitoring and Diagnostics with Intelligent Agents using Fuzzy Logic," *Engineering Letters,* 2007.

[70] Mario Gongora, "Adaptive Intelligent Agents based on Efficient Behaviour Differentiation Models," in *IEEE ANDERSCON*, Bogota, 2010.

[71] Umair Rafique, "Motivation Based Goal Adoption for Autonomous Intelligent Agents," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Lyon, 2011.

[72] Amanda Coles, "Forward-Chaining Partial ORder Planning," 2010.

[73] F. Ngemoh, Modeling the Automated Screw Insertion Process, PhD Thesis, London: King's College London, 1997.

[74] Sensoray, "Sensoray Model 626," Sensoray, [Online]. Available: http://www.sensoray.com/products/626.htm. [Accessed 10 2013].

[75] Mecsense, "Mecsense Kraftmesstechnik," Mecsense , 2012. [Online]. Available: www.mecsense.de/Ringkraftaufnehmer/Kraftmessring-Mecsenseflach/KMR-50kN-0-5.html. [Accessed 2012].

[76] G. Dinger, C. Friedrich, "Avoiding self-loosening failure of bolted joints with numerical assesment of local contact state," *Engineering Failure Analysis,* vol. 18, pp. 2188-220, 2011.

[77] X.-G. Duan, "A Saturation-based Tuning Method for Fuzzy PID Controller," *IEEE Transactions on Industrial Electronics,* vol. 80, no. 11, pp. 5177-5185, 2013.

[78] H. H. Choi, "Design and Implementation of a Takagi-Suegeno Fuzzy Speed Regulattor for a Permanent Magnet Synchronus Motor," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 8, pp. 3069-3077, 2012.

[79] C. Westermayer, R. Priesner, M. Kozek, R. Bauer, "High Dynamic Torque Control for Industrial Engine Test Beds," *IEEE Journal of Industrial Electronics,* vol. 60, no. 9, pp. 3877 - 3888, 2013.

[80] Alessando Pisano, "Cascade Control of PM DC Drives Via Second-Order Sliding-Mode Technique," *IEEE Transactions on Industrial Electronics,* vol. 55, no. 11, pp. 3846-3854, 2008.

[81] Baeksuk Chu, "Mechanism and Analysis of a Robotic Bolting Device for steeel beam assembly," in *International Conference on Control, Automation and Systems*, Korea, 2010.

[82] G.M. Joselin Herbert, "A review of wind energy technologies," *Renewable & Substainable Energy Reviews,* vol. 11, pp. 1117-1145, 2005.

[83] A. Mittal, "Genetic Algorithm based Tuning of fixed biased PID controller for a Nonlinar Constant Temperature Water Bath unter Load Disturbances," *Automation and System Engineering,* vol. 6, no. 2, pp. 96-109, 2012.

[84] Sayeed Mir, "PI Fuzzy Estimators for Tunig the Stator Resistance in Direct Torque Control of Induction Machines," *IEEE Transactions on Industrial Electronics,* vol. 13, no. 2, pp. 279-287, 1998.

[85] Sayed A. Nassar, "Torque-Angle Formulation of Threaded Fastener Thightening," *Journal of Mechanical Design,* Vols. 130 / 024501-1, 2008.

[86] F. Mrad, "Fuzzy Logic Control of Automated Screw fastening," in *Industry Applications Conference* , Orlando, FL, 1995.

[87] Z. Michalewicz, Genetic Algorithm + Data Structures = Evolution Programs, New York: Springer Verlag, 1994.

[88] Michael Wooldridge, "Intelligent agents: theory and practice," *The Knowledge Engineering Review,* vol. 10, no. 2, pp. 115-152, 1995.

[89] Marco Siserre, "Future Energy Systems," *IEEE Industrial Electronics Magazine,* 2010.

[90] M. Sugeno, K. Tanaka, "Successive Identification of a fuzzy model and its application to prediction of a complex system," *Fuzzy sets and systems,* vol. 42, pp. 315-334, 1991.

[91] F. N. S. E. K. A. L.D. Seneviratne, "Theoretical modelling of the self-tapping screw fastening process," *Journal of Mechanical Engineering Science,* vol. 215, no. 2, pp. 135-154, 2001.

[92] P. Hajek, "Fuzzy Logic," Stanford, [Online]. Available: http://plato.stanford.edu/entries/logic-fuzzy/. [Accessed 13 06 2012].

[93] A. Pisano, L. Fridman, E. Usai, "Cascade Control of PM DC Drives Via Second-Order Sliding Mode Technique," *IEEE Transactions on Industrial Electronics,* vol. 11, no. 55, pp. 3846-3854, 2008.

[94] Z. Hinhua, "Self-Organizing genetic algorithm based tuning of PID controllers," *Information Sciences,* pp. 1007-1018, 2009.

[95] Yi Wan, "A new self-adaptive control model and application basis on optimum Fuzzy RBFNN," in *International Conference on Artifical Intelligence and Computational Intelligence*, Shanghai, 2009.

[96] Xiwen Zhang, "An Improved Torque Method for Preload Control in Precision Assembly of Miniature Bolt Joints," *Journal of Mechanical Engineering,* vol. 58, no. 10, pp. 578-586, 2012.

[97] XinJiang Ku, "Nonlinear measurement based on Integrated Robust Design and Control for Manufacturing System," *IEEE Transactions on Industrial Electronics,* vol. 60, no. 7, pp. 2711-2720, 2011.

[98] W. Zhixin, J. Chuanwen, Ai Qian, Wang Chengmin, "The key technology of offshore wind farm and its new development in China," *Renewable Energy,* vol. 13, no. 13, pp. 216-222, 2009.

[99] Toshiya Ueno, "Establishment Of Bolt Tightening Simulation System For Automotive Industry Application Of The Highly Reliable CAE Model," *International Business & Economics Research Jorunal,* vol. 8, no. 5, pp. 57-67, 2009.

[100] T. Fujinaka, H. Nakano, S. Omatu, "Bolt tightening control using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 3, pp. 1390-1395, 2001.

[101] R. F. Stengel, "Intelligent Failure-Tolerant Control," *IEEE Control Systems,* pp. 14-23, 1991.

[102] S. Srikanth, "Modeling and PID Control of the brushless DC Motor with the help of Genetic algorithm," in *International Conference on Advances In Engineering, Science and Management*, 2012.

[103] R. Vinodihini, "Genetic Algorithm optimized on-line Neuro-tuned robost position control of BLDC Motor," in *IEEE Conference on Electrical, Electronics and Computer Science*, 2012.

[104] Pawel Z. Grabowski, "A Simple Direct-Torque Neuro-Fuzzy Control of PWM-Inverter-Fed Induction Motor Drive," *IEEE Transactions on Industrial Electronics,* vol. 47, no. 4, pp. 863-870, 2000.

[105] M. McKimm, "BBC," BBC, 6 6 2012. [Online]. Available: http://www.bbc.co.uk/news/uk-northern-ireland-18330468. [Accessed 11 2013].

[106] M. Klingajay, L.D. Seneviratne, K. Althoefer, "Identification of threaded fastening parameters using the Newton Raphson Method," in *Proc. of the International Confernce on Intelligent Robots and Stystems (IROS)*, 2003.

[107] Loic Michel, "Model-free control of dc/dc converters," in *Workshop on Control and Modeling for Power Electronics*, Boulder, CO, 2010.

[108] James Carvajal, "Fuzzy PID Controller: Design, performance evaluation and stability analysis," *Information Sciences,* pp. 249-270, 2000.

[109] Ibermatica, "COSMOS," Ibermatica, 2011. [Online]. Available: http://www.cosmosproject.eu/. [Accessed 01 01 2013].

[110] Humberto Henao, "Wire Rope Fault Detection in a Hoisting Winch System by Motor Torque and Current Signature Analysis," *IEEE Transactions on Industrial Electronics,* vol. 58, no. 5, pp. 1727-1736, 2011.

[111] Hiroki Yamada, "Highly-Reliable CAE Analysis Approach - Application on Automotive Bolt Analysis".*System Science and Simulation in Engineering.*

[112] G. J. Herbert, "A review of wind energy technologies," *Renewable and Substainable Energy reviews,* vol. 11, pp. 1117-1145, 2007.

[113] H.V.H. Ayala, "Tuning of PID controller based on a multi objective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications,* pp. 8968-8974, 2012.

[114] M. Narimani, H.K. Lam, "Quadratic stability analysis of fuzzy model based control systems using staircaise membership functions," *IEEE Trans. Fuzzy Systems,* vol. 18, no. 1, pp. 125-137, 2010.

[115] A. Malki, H. Li, G. Chen, "New design and stability analysis of fuzzy proportional-derivative control system," *IEEE Trans. Fuzzy Systems,* vol. 2, no. 4, pp. 245-254, 1994.

[116] H. Li, "Overview of different wind generatior systems and their comparisons," *IET Renewable Power generation,* vol. 2, no. 2, pp. 123-138, 2008.

[117] H. Li, "Design optimization and site matching of direct drive wind power generator systems," *Renewable Energy,* vol. 34, no. 34, pp. 1175-1184, 2009.

[118] H. K. Lam, "Stability analysis of fuzzy-model-based control systems: application on regulation of switching DC-DC converter," *IET Control Theory and Applications,* vol. 3, no. 8, pp. 1093-1106, 2009.

[119] E. Kamal, A. Aitouche, R. Ghorbani, "Fuzzy Scheduler Fault-Tolerant Control for Wind Energy Conversion Systems," *IEEE Transactions on Control Systems Technology,* 2013.

[120] H.K.. Lam et al, "Control Design for Interval Type-2 Fuzzy system under Imperfect Premise Matching," *IEEE Transactions on Industrial Electronics,* 2013.

[121] M. Narimani, H. K. Lam, K. Althoefer, R. Dilmaghani, C. Wolfe, C. Deters, "An Approach for Stability Analysis of Polynomial Fuzzy Model-Based Control Systems," in *IEEE International Conference on Fuzzy Systems*, Sydney, 2011.

[122] R. Technologies, "PCB Loard Torque," 24 06 2014. [Online]. Available: http://www.pcbloadtorque.com/pdfs/engineering%20fundamentals.pdf.